

M.5529 Oracle/Sun Merger
Monty Program Ab's observations and corrections
on Prof. Eben Moglen's submission to the Commission
regarding GPL, forking and MySQL business models.

1 Background and introduction

1.1 A lawyer well-known in and well-respected by the free software community, former legal counsel to the Free Software Foundation, and former advisor to MySQL Ab, Professor Eben Moglen on December 4 published an opinion paper dated and submitted to the European Commission on November 19.¹

1.2 Monty Program Ab was founded in 2009 by the creator of MySQL and its dual license business model, Michael "Monty" Widenius, and employs some former employees of MySQL Ab and Sun Microsystems. Like our previous submissions in this case, we wish to provide here our unique insight into MySQL, be it from a technical, business, or sometimes even licensing perspective.

1.3 In the following paragraphs we will correct (only) the main factual errors of Prof. Moglen's paper. Before commencing it is important to stress our highest regard for the valuable, life-long and groundbreaking work of Prof. Moglen for the FSF, MySQL Ab, and software freedom in general. We would like to remind the reader that in proceedings like this, each party is always advised and supported by lawyers, and a lawyer's duty is to argue that party's case as well as possible. Hence, with our pointing out the errors in Prof. Moglen's paper, it is not our intention to imply to any reader anything about the person of Prof. Moglen himself and we strongly advice the reader to read the following with a similar attitude.

1.4 Monty Program Ab is at this stage bound by strict confidentiality regarding the content of the European Commission's Statement of Objections, and unlike Oracle, we cannot in this paper debate any references to the SO, nor whether or how much the opinion paper is even relevant to the Commission's investigation. We have similarly avoided addressing a part of Prof. Moglen's paper that we believe discuss business secrets of Oracle. However, we will still be more than able to correct the errors in Moglen's text itself by publicly known facts.

1.5 Both MySQL AB and Sun Microsystems Inc commonly use the term "*open source*" to describe the kind of software that MySQL also is, and the Commission has also done so in this investigation. Out of respect for both Prof. Moglen, and particularly the inventor of the GPL license itself, Dr. Richard Stallman, we will in this paper use the historically predating term "*free software*" and purely as synonymous to "open source". It should also be clarified to any uninitiated reader that this term does not indicate whether MySQL or another piece of software is made available gratis or against a fee, both can be and are instances of free software.

1.6 Numbers in parenthesis are references to Prof. Moglen's position paper. (see footnote 1)

2 General comment on the relevance of debating forking

2.1 Prof. Moglen's paper focuses on MySQL's licensing and business models, and whether someone could or even should fork MySQL.

2.2 Forking is an important safeguard provided by all free software licenses. But it is important to remind, that forking is considered as undesirable and something usually done as a last

¹ <http://www.softwarefreedom.org/news/2009/dec/04/software-freedom-law-center-submits-opinion-oracle/>
<http://www.softwarefreedom.org/resources/2009/Oracle-Sun-EC-opinion.pdf>

resort. It should be widely accepted that forking in short term will be harmful to a project's ecosystem by dividing it into two (or more), and sometimes ends up being harmful to the project even on a long term basis (such as the BSD operating system, which has split into 4-5 forks).

2.3 Whether it would be feasible or required to do a fork of MySQL or not, the fact that it is a focus in this debate is in itself alarming and validates the Commission's concern for MySQL. For instance, when MySQL was acquired by Sun, there was no speculation for a need of forking MySQL whatsoever.

3 Comparisons to other free software applications, libraries, infrastructure components also licensed under the GPL (§§11, 14, 15, 19, 29, 43, 48)

3.1 Prof. Moglen describes MySQL's single vendor controlled development model of a GPL licensed product as atypical, whereas other GPL projects like Linux are labeled typical. While this characterization may be justifiable from a certain ideological perspective, we should note that MySQL's (and Trolltech's) successful execution (both commercially and from a product adoption point of view) of the single vendor and dual license development model has in fact led to an increased popularity towards the model. Since Prof. Moglen seems to either dislike or regard the model as categorically inferior, we may as well point out Richard Stallman's recent strong endorsement of it at the FOSS.my conference.²

3.2 While Prof. Moglen must be intimately familiar with it, he fails to mention the tactical way license choices are made and have been made by the FSF and its President Richard Stallman. (Indeed, at the FOSS.my conference Stallman confesses to often having advocated the use of so called "permissive" licenses, mentioning in particular the OGG audio/video codec libraries.) The Free Software Foundation is the issuer and main promoter of the GPL and itself the copyright holder of some of the most important free software in the world.

3.3 The FSF strategy seems to be to maximize the success of free software by appropriately balancing the use of

- the GPL license, which has a strong reciprocity requirement for any other software "touching" the GPL software, thus providing strong "leverage" to compete against proprietary software.
- the LGPL license (*Lesser GPL*, formerly *Library GPL*) which has a reciprocity requirement for changes and enhancements to the software module itself, but not for independent modules that utilize it. ("touch" it, for lack of a better word.)
- at times, when appropriate for instance to maximize interoperability, even simpler, so called "permissive" licenses

3.4 In particular, with only minor exceptions (notably *readline*, a library also used in MySQL), we observe the FSF using the GPL only for standalone applications or utilities, such as Emacs or GCC.

3.5 We observe the FSF using the LGPL for libraries, with the explicit intent that also non-GPL applications may therefore use them. The underlying theory is that a common library is best used as exactly that - commonly by all developers and all software under various licenses.

3.6 For completeness, we observe the FSF not widely having adopted the AGPL license, with an even stronger reciprocity requirement than GPL. Also known as the "GPL for web services".

3.7 Hence, if we imagine for a moment that MySQL had been developed as an FSF project, we can say almost with full certainty that it would have been licensed under the LGPL, since its primary use is by applications utilizing it as their data store either by embedding or connecting to a MySQL database using and embedding one of the MySQL connectors (aka. drivers, client

² <http://www.bytebot.net/blog/archives/2009/10/25/brian-aker-debates-with-richard-stallman>

libraries). The reason MySQL chose the GPL is of course exactly to exclude a class of software - namely proprietary software - from using it for free, so that such developers must instead pay to obtain a non-GPL copy of MySQL. This is the dual licensing business model.

3.8 However, we should immediately note that LGPL would be an awkward license choice (to say the least) for MySQL, due to it explicitly defining "The Library" as the free software part and "The Application" as the proprietary part, which would be backwards for all MySQL plugins, in particular the storage engines. In addition there are probably some other corner cases of MySQL OEM customers who could not use an LGPL license to pursue a similar strategy that they can today with their license from MySQL.

3.9 This brings us now to the FOSS exception Prof. Moglen discusses in (§29). The effect of licensing MySQL under the GPL, had the unintended consequence of other free software projects licensed under other licenses (and some that had significant value to MySQL, from a distribution and adoption point of view, such as PHP, Perl) feeling they could no longer distribute the MySQL client library. To remedy this, MySQL issued a FOSS exception to enable that. In essence, MySQL's strategy was to allow free use for all open source applications, while selling non-GPL licenses to proprietary applications. (For completeness, in-house software as well as web services are not distributed from one party to another, so they are not affected by the reciprocity requirements of the GPL and typically would not acquire the non-GPL license.)

3.10 For reasons unknown to us, the FOSS exceptions list does not today include GPLv3! Since MySQL is licensed under GPLv2, and these licenses are not compatible, this means that today any GPLv3 application cannot legally use MySQL (at all, since due to the GPLv3 requirements it also couldn't use the commercially licensed version of MySQL even if willing to pay). This means that by owning the copyright to MySQL, Oracle is in a position to not only harm the proprietary applications that are customers to MySQL, but even part of the free software community. Given that GPLv3 is expected to become possibly the most popular free software license, this is a significant part of the free software community. Even if we understand Prof. Moglen is arguing a specific side of this debate, we would have expected him to mention this significant omission of the license he himself is most familiar with.

3.11 Throughout the document Prof. Moglen refers to various GPL licensed free software projects as models that MySQL itself or at least a MySQL fork should adopt. None of these actually are good comparisons to MySQL.

3.12 Linux is commonly seen as the most successful GPL software in the world, both commercially and non-commercially. But to be precise, Linux is actually licensed with a "GPL+exception"³ license that makes it clear that the GPL is only applied to the Linux kernel itself, while software running on top of the operating system can be licensed non-GPL, including proprietary.

3.13 First of all, OpenOffice is licensed under the LGPL and not the GPL. This makes all the difference if discussing forking of MySQL, since if MySQL was licensed under the LGPL most of the obstacles related to a GPL only fork would immediately be resolved (with the important reservations we have noted above in 3.8).

3.14 Also, OpenOffice is an application, never embedded into other applications, so even if it were licensed under the GPL, it would be a poor example when discussing the forking of MySQL.

3.15 The GCC processor is also an application or perhaps a tool. It is possible to use GCC to compile any source code into a binary program, without the resulting program therefore becoming GPL licensed. This means that it makes sense for a proprietary vendor to invest in a fork and extend GCC to get better performance for their application/hardware as their application that is compiled by GCC is not affected by the GPL.

3 If some would prefer to call the exception as merely a clarification, the effect is the same and MySQL specifically has denied the applicability of the same principle to MySQL.

3.16 Emacs and Xemacs are applications. (To the extent they are used for programming of software, the source code files written in Emacs/Xemacs do not need to be licensed under the GPL.)

3.17 Emacs was forked by Lucid, who created Xemacs because of their need to use it as an editor in their Integrated Development Environment. The other components of the IDE suite were however not GPL, but proprietary, since the GPL license of XEmacs in no way extends to them. Thus the financial incentive to invest in developing XEmacs - on a GPL basis - was purely to use it as a component in a proprietary package.⁴

3.18 Samba at least has one thing in common with MySQL in that it is a client-server architecture. However, it provides network file sharing over the standard CIFS protocol (same as the proprietary Microsoft Windows) and it is commonly accepted that the GPL cannot transfer its reciprocity requirements over such a standard interface. (Prof. Moglen agrees in §21.) Thus, an application or operating system can utilize a Samba file system without having to be under the GPL license. In particular, the common use case for an application storing files on a CIFS network share is to do so by normal operating system calls to the operating system that has mounted the share, which is why most applications "using" Samba are completely unaffected by its GPL license.

3.19 Samba also provides a client library that is GPL licensed. This is similar to MySQL's client libraries being GPL licensed. In both cases only GPL licensed applications and operating systems can utilize these libraries to connect to any CIFS fileshare (not just Samba) and a MySQL server respectively. However, a core difference is that there exist also non-GPL libraries that can connect to CIFS fileshares, whereas MySQL's own GPL libraries are the only ones available that fully implement the proprietary MySQL client-server protocol. Hence, it is possible to build also non-GPL applications that can connect to CIFS fileshares (Samba, or otherwise), while it is not possible to do the same for MySQL.

3.20 There is legal uncertainty, which is significant due to the incentive of MySQL's sales force and legal counsel to interpret this in a way commercially beneficial to the MySQL dual licensing business, whether creating a non-GPL MySQL client library would even suffice to circumvent the GPL requirement; and if so, in which cases specifically. This is due to the MySQL client-server protocol being particular to MySQL and not a standard protocol like CIFS. In any case it is well established that any (distributed) application that requires MySQL to work and especially if it's distributed with MySQL is always affected by the GPL license.

3.21 Mambo has a similarity with MySQL in being a single vendor GPL product, with a dual licensing business model targeting proprietary plugin functionality and themes. While Joomla has succeeded as a fork, it also proves the problem being discussed for MySQL, as these proprietary Mambo extensions now cannot be utilized in the GPL-only Joomla fork. There currently exists some gray market activity where proprietary Joomla plugins and themes are being circulated anyway, mostly since the copyright holder of Mambo does not have the resources to actively assert its rights against these, and the copyright holders of Joomla don't have the motivation or resources. We assume that Prof. Moglen didn't intend to suggest this kind of gray market as a remedy against anti-competitive behavior from Oracle as an owner of MySQL. (See also GPL case law section below.)

3.22 Mambo and Joomla are Content Management Systems that are only used to create web applications, either for the Internet or corporate intranets. Since these are often custom and in-house applications, the need to distribute them is not high and thus the GPL doesn't affect them in any way. This is not the case for MySQL which is used across the software industry across all kinds of applications. Indeed, for the web services using MySQL, a GPL only fork (if viable) could from a strictly legal perspective serve as a replacement to MySQL, but unfortunately this then leaves the rest of MySQL's users and use cases without remedy. (We separately question the

⁴ <http://www.dreamsongs.com/IHE/IHE-74.html>

viability of a supposedly general purpose RDBMS project that addresses only some but not all of the market, due to legal restrictions.)

3.23 The above overview of Prof. Moglen's examples shows that none of them is comparable to MySQL, due to having various exceptions to the GPL that MySQL doesn't have, due to not being "infrastructure" software but rather standalone applications, or in the case of Joomla simply showing a bad example to where such a fork can lead. As Prof. Moglen is one of the authorities on free software and should thus know of all the important forks in the GPL history, we have to assume by the examples he gives of forks that there exist no good examples of a successful fork (by adoption, but also economically) of a piece of infrastructure software like MySQL.

3.24 Instead the above analysis of both FSF's software and Prof. Moglen's examples proves that a successful strategy for any free software project is often to at least choose a license that enables maximum adoption and applicability, rather than constrain its potential audience by legal grounds. Sometimes this is by adding exceptions to the GPL, sometimes using LGPL, and sometimes other permissive licenses.

3.25 The discussion in (§15) is whether big corporations rather contribute to ASL or GPL licensed projects is therefore fundamentally flawed, if one accepts the above evidence that corporations contribute to projects they can use widely for many purposes and are unlikely to contribute to projects with narrower applicability - such as a web only database. For instance, it is undoubtedly the case that Oracle's main motivation to contribute to Linux is certainly because it can run its proprietary database and middleware on Linux servers.

3.26 Given all of the above, Prof. Moglen's advice in (§48) that Oracle should incorporate GPL code from third parties without copyright assignment is extremely dangerous, and we have to admit it is a threat scenario we previously have failed to identify to the Commission. Such a strategy would allow Oracle to immediately make MySQL unusable to the current customers to the non-GPL license, and by cascading effects also to those end users that would want to use MySQL as an efficient and affordable database for such OEM and ISV applications. In addition this practice would void the current MySQL FOSS exception for free software under other licenses than GPLv2 and would make MySQL unavailable for them too. It is ironic that Oracle could now pursue this extremely harmful strategy with the blessing from the SFLC.

3.27 At this point it is appropriate to finally mention, that Prof. Moglen's paper focuses solely on the GPL based usage of MySQL, as if the commercial license customers didn't even exist. It is a mystery to us if this is a fundamental misunderstanding of the MySQL business model (given his background, highly unlikely), or a tactical omission to simply not mention the most problematic area of this investigation.

3.28 In the unlikely case that Prof. Moglen's (poorly) expressed or hidden intent is that all of that software that currently uses MySQL under the commercial non-GPL license (aka OEM license), could somehow all be forced to become GPL licensed free software by this manoeuvre, we must note that we don't see that as a realistic outcome, nor do we think pursuing such an agenda is within the jurisdiction of the Commission. The OEM license version of MySQL today powers such things as our GSM network infrastructure (Alcatel-Lucent, Nokia Siemens Networks, Ericsson), Internet infrastructure (Cisco), some unnamed Adobe applications, just to start and not to forget the storage engines we'll discuss later. As much as we too would like to see this vast software property to all become free software, we feel this agenda would be dangerously misguided as an argument in this case.

4 Comparisons to occurrences of re-branding of free software programs or forks (§35)

- 4.1 Prof. Moglen's examples of re-branding of some free software applications are correct, but very poor analogies for the situation a MySQL fork would be in.
- 4.2 GAIM/Pidgin is not a fork, rather the application changed its name, replacing the old name, and continuing with the same developer team, distribution channels and user base.
- 4.3 Firefox is technically a fork of the old Mozilla code base, but happened with the support of the Mozilla project and foundation. The full name is still today "Mozilla Firefox".
- 4.4 Iceweasel is not a fork in the sense that the Debian team doesn't actively develop and innovate the browser beyond changing the name of the application. In addition, it is merely a replacement of Firefox within Debian, with no ambitions to actually compete against Firefox outside of Debian.
- 4.5 While not fundamentally flawed examples as such, we find all of the above as poor ways to estimate the re-branding and trademark challenges faced by a MySQL fork.

5 Financial incentives to develop MySQL or a fork

- 5.1 Throughout the paper (such as §44-45) Prof. Moglen proposes new business models for MySQL and asserts that they would be better than the ones MySQL has historically chosen to use. With our unique experience of the whole history of MySQL, we are confident we are in a better position to comment on these issues than Prof. Moglen is. The same is true for his opinions on technical matters in the section following after this one
- 5.2 It is an undisputed fact, that second to Red Hat (or Linux, if rather taken as a whole) MySQL is to date the commercially most successful free software business . It would be much more reassuring to see real examples that are relevant to MySQL, than suggestions based on wishful thinking. MySQL has been available as GPL this whole decade, so one would expect if commercially more lucrative and efficient models to develop it on a purely GPL basis exist, other companies large and small would have rushed to the opportunity to do so.
- 5.3 Similarly, as Prof. Moglen agrees, it is a fact that MySQL was almost fully developed by employees of MySQL Ab and later Sun's MySQL division. This work was funded by the dual licensing revenue and subscription (which includes proprietary software and services) revenue from MySQL customers, all of which is a prerogative of the copyright holder and not available to a fork. We do not deny that other free software projects have other development models than MySQL. At the same time, historical evidence is what it is, and Prof. Moglen resorts to unproven speculation when claiming to know of better models for MySQL.
- 5.4 Of the current so-called forks of MySQL, none is actually independent of the official MySQL. MariaDB, Percona builds and OurDelta all continue to "merge" bug fixes from new MySQL releases, so Sun's contribution to them far outweighs their own specific enhancements. There is no evidence to assume they would be viable independently of the existence of Sun's MySQL. The fourth fork we are aware of, Drizzle, has indeed broken with the MySQL code base. Drizzle's main funding comes from Sun itself, it is still far from being a production quality project, and even then it's goal is explicitly not to be a replacement for MySQL. Drizzle only targets the cloud use case, but none of the traditional uses of MySQL and could not for technical reasons be a replacement.
- 5.5 Despite all of the above, even if there was a viable business model that could finance the development of a fork (by one or many companies), this still leaves the current customers of the

non-GPL MySQL version without any help. In addition, a fork could not provide the services (support, consulting, training, marketing) and reassurances (like indemnification and certification) that large enterprise customer would typically require to be able to use the fork.

5.6 Given the previous section, we strongly doubt whether a MySQL fork could then be successful even in the remaining market. For instance, with the experience we have in selling databases to enterprise end users, we suspect they would be unlikely to be interested in an RDBMS solution that they could use for their in-house projects, but not for 3rd party software. Typically end users tend to have a strategy of consolidating their database use to one or a few vendors for all their applications, not choosing databases that could only be used for a subset of applications. As a result, it is likely that a GPL-only fork would not be viable for a much larger segment of the database market than purely OEM customers, and then long term not viable at all.

5.7 Finally, if it were easy or feasible to fork MySQL, Sun wouldn't have needed to pay 1 billion USD to acquire MySQL AB, nor would there be any need for Oracle (other than malign intentions) to desperately hold on to it, since it would have been much cheaper to fork the source code instead.

6 MySQL storage engines, design decisions, Falcon, etc... (§18)

6.1 While this paper does not address point by point all paragraphs in Prof. Moglen's paper, paragraph 18 is so filled with substantial and significant errors it makes sense to devote a full section solely to clarify and correct.

6.2 Upon request by the Commission, Monty Program has previously submitted to the Commission a brief overview of existing and active 3rd party storage engines we are aware of. It seems that

- 1 is dual licensed (like MySQL)
- 2 are "open-core", ie a GPL version exists but a premium version is proprietary
- 5 are fully proprietary
- 6 are GPL only. However most (it seems all) of these GPL-only projects are hobby projects of 1 or 2 individuals or at best in-house projects of one company, indeed, half of them are free-time projects of Sun employees.

(the list excludes InnoDB and most likely a number of lesser known hobby projects)

Hence, while it is true that *hypothetically* 3rd parties could develop storage engines under any licensing scheme and with or without VC funding, *reality* shows all of the ones with commercial ambitions (such as to offer support and compete with Oracle's database offerings) are dependent on the commercial non-GPL license for MySQL for their chosen strategy, and have received funding from investors.

6.3 *"This, I think, is a misunderstanding of the technical history. MySQL's design was an outcome of the need for dual-license revenue, not a pre-condition to it."* This is not true. MySQL's storage engine architecture is a result of its technical evolution, where it from the early days had to support several different formats of storing data, which was enabled by using different storage engines from different vendors, because of customer demand. And this architecture predates the dual-licensing business model by a decade. See also 6.5.

6.4 *"PostgreSQL, for example, does not have or need multiple storage engines: it contains a single highly-configurable storage manager which is not independent of the database engine."* The first part is correct, but the implication that the PostgreSQL architecture is therefore superior is not true, at best it is an opinion here stated by a lawyer. On the contrary, modularity and a "right tool for the job" attitude is commonly considered a virtuous principle for software design and among MySQL users considered as a unique strength to MySQL.

6.5 *"MySQL's use of multiple storage engines resulted from MySQL's need to have something to sell dual-license customers different and better than that provided to users under GPL."* This is

not true and evidenced from the fact that Ingres has the same business model as MySQL, but does not have a modular storage engine architecture. Also, the MySQL server sold by MySQL AB under the non-GPL license was always identical to the one provided under GPL. Possibly Prof. Moglen has not understood that most dual license customers are application vendors that use MySQL (with any of the available storage engines) as their database.

6.6 *"The resulting uncertainty that could arise over whether a table had been created using a transaction-safe storage engine or by the original freely-available MyISAM storage engine contributed to the problem with occasional corruption of MySQL databases that was long a primary drawback to the enterprise use of MySQL."* The sentence would be correct if it said "long ago", as this was indeed the case about 10 years ago.

6.7 *"MySQL AB contemplated and designed a single multi-purpose high-performance storage engine, named Falcon, that would have substantially replaced all existing MySQL storage engines,"* Falcon was intended to complement and add to the diversity of existing storage engines as can be verified on any MySQL user conference presentation on this topic and certainly could not have functionally replaced most of the other MySQL engines. The major reason for the Falcon initiative was Oracle's acquisition of InnoDB - which is relevant to point out in this investigation.

6.8 *"but the business-model consequences of that step were sufficiently negative for MySQL AB, because of its need to produce dual-license revenue, that the project was never completed."* At this point we are not even sure what Prof. Moglen is trying to say anymore, but it is certainly not correct in any case. Falcon would not have (had) any particular effects on MySQL's business model(s) or ability to produce dual-license revenue, other than providing more independence from Oracle as an important subcontractor.

7 Various corrections

7.1 We disagree that MySQL wouldn't today have a very vibrant ecology around it (§19). However, regarding code contributions to MySQL itself, Mr Widenius is known to agree that external contributions did not happen as often as he himself would have wished and he was actively trying to remove obstacles to such contributions. However, we disagree that the primary reason for this would have been the need for copyright assignments and MySQL's dual licensing business model. As evidence we present the fact that FSF also requires copyright assignment for all of its projects, and never had any difficulty whatsoever to receive millions of lines of code contributions. Clearly there have been numerous other things MySQL Ab managed poorly with regard to fostering a strong community of code contributions, not the least of which was the fact that when such contributions happened, and the contributors were willing to sign a copyright assignment, MySQL Ab was still slow to receive those!

7.2 In (§30) Prof. Moglen again gives his assessment on the technical superiority of Oracle over Microsoft SQL Server and MySQL, which again, at best is a lawyers opinion.

7.3 (§45) There are no 2 separate MySQL source trees, the dual licensed MySQL server is the same as the GPL licensed.

7.4 It is far beyond our competence to discuss software patents, except that it is our firm belief the whole system of software patents is harmful to forks, MySQL itself and all software development. However, we cannot help but getting the impression that Prof. Moglen in (§37) cites a ridiculously narrow case of verbatim copying of pre-existing source code from one place to another, and even then his comment doesn't fully assure us that even that would be safe from a software patent aggression perspective! This more than well illustrates the uncertainty a software developer faces with regard to software patents.

7.5 We are almost sure that the Commission is not concerned over the availability of historical versions of MySQL, which of course are irrevocably under the GPL, but rather interested in

ensuring that also a new owner continues to make it available under similar terms that Sun has done, providing continuous bug fixes, enhancements and innovation, under the free software GPL license and also available under dual licensing for proprietary software and the FOSS exception for free software under other licenses. (§38)

8 How a storage engine hypothetically could circumvent the GPL requirements (§21), and how a MySQL based application also couldn't (n/a)

8.1 Prof. Moglen lays out a hypothetical scenario where a storage engine provider could distribute a proprietary engine that is claimed to be useful either standalone or if plugged into some non-GPL database, and it could be used also with MySQL since it (by co-incidence, almost) also works as a MySQL storage engine.

8.2 The MySQL storage engine architecture is unique and even a remotely similar one does not exist in any other commonly used RDBMS. So the second option is purely hypothetical.

8.3 The MySQL storage engine architecture is highly specific to MySQL and involves manipulating internal data structures that date back to the product's birth in the 1980's (then known as *Unireg*). It would not be credible to claim that a standalone engine happens to implement this particular interface "by accident".

8.4 Even if Prof. Moglen's hypothetical situation were reality, it would be a legally insecure basis to build a business on, or at best an awfully awkward way of distributing to customers via separate channels the pieces of a product.

8.5 Prof. Moglen's paper again fails to address the main area for concern, all the applications using MySQL as an RDBMS under the commercial non-GPL licenses, meaning, there is no advice how they could possibly circumvent the GPL requirements if needed. We note that similar problems as we have here pointed out for storage engines would arise also for applications.

9 GPL case law and NuSphere (§23-25)

9.1 Prof. Moglen spends some effort in explaining how complex the commercial relationship and following dispute was between MySQL Ab and NuSphere. After this he neglects to dispute the Commission's conclusion that NuSphere by distributing a proprietary closed source storage engine together with the GPL version of MySQL, was in violation of the GPL and was forced to remedy this by releasing the source code to the proprietary engine as GPL. This is because, he of course could not deny that.

9.2 Indeed, it is very much thanks to Prof. Moglen's own work both in that case and more widely in defending and asserting the GPL in various cases, not the least as part of his valuable work as a former counsel for the Free Software Foundation, that there today is no doubt about the validity of the GPL as a strong and sound free software license, and that any attempts to circumvent its requirements are not supported by case law. It is in fact appropriate to end disagreements on the assessment of the proposed takeover of Sun by Oracle aside, by again reiterating our gratitude towards this life-long work of Prof. Moglen.