Question #1

Please evaluate Microsoft's offer to provide a source code licence to undertakings having entered into a WSPP agreement from your company's technical and legal perspective.

FSFE and the Samba Team take the view that the offer of Microsoft to license for reference purposes only the source code of part of the Windows operating system is **grossly inadequate** to fulfill its obligations arising from the Decision. **By no means** we think that this offer can **replace or reduce the burden to prepare sufficiently detailed, correct and comprehensive information**. It could be a useful source of fine grained information that have already been displayed on general grounds in the documentation, or to cross check possibly erroneous or incomplete information, but the two kind of documents are at entirely different levels. As we have publicly said, it is like one party is requested to produce a map of a city, and in reply it provides the blueprints of any single building.

Moreover, we see no guarantee that the licensed source code would be the very same as that used internally by Microsoft, *e.g.*, with the same writing quality (such as proper indentation) and with the same – or better – comments. Without these requirements the offer would be absolutely meaningless.

All in all, while it is unprecedented that Microsoft offers inspection of its source code to competitors, the **main areas of concern** of FSFE and Samba Team are **still untouched** by this offer. Therefore, there is little to comment on the offer itself, and the general inadequacy of the compliance scheme envisaged by Microsoft depends both on the irrelevance of the offer and on the legal grounds which this offer shares with the entire WSPP compliance offer, which has already commented at length in our previous submissions.

Legal issues

From the point of view of Free Software, and as we more clearly state below, the offer of Microsoft does not improve by any means the quality of compliance. Free Software operators are **completely barred** from using the compliance scheme, they are equally barred from using this accessory to the compliance scheme. It is like a train that does not stop at your station: it does not change anything if it has a new and very comfortable wagon.

If the train will eventually stop at the Free Software station, however, the additional wagon would be as well useless. More details are given under answer to question 2.

The concerns of FSFE and the Samba Team can be summed up with one single sentence: that the Source code reference license is a **poisoned honeypot**, from which Free Software operators shall stay away as much as possible. Actually seeing source code being not under a Free Software license (such as the BSD, the GNU LGPL, or even the MicroSoft Public License) means to expose any Free Software project, including Samba, to huge **litigation risks**. Inspection of the “licensed” source code by one member of a group would mean that this person would have enormous obligations to check that any additions, patch, forking of its project would never ever include any copyrighted relevant sequence of instructions made by anybody.

Chances are considerably high that others who contributed to the project, by limited coincidence (software programming languages are more formally constrained than natural languages), would sometimes end up with the same coding sequence while addressing the same problem as the “licensed” code. If this person has not seen the code, he can at least claim good faith and design/language constraint. But if a member of the group has seen the same code, apart the “residual” knowledge defence, there would be a high risk of litigation on a trade secret infringement basis too, which cannot be resolved by a simple and prompt change of the infringing code (as Microsoft's lawyers cleverly said “*once the bell has rung, it cannot be unrung*”).

This risk is commonly referred to as “**code poisoning**”, being this “true” or just “alleged”.

Technical issues

Not necessarily the source code is useful for reference purposes. Most of the code reflects particular underlying choices, which for several reasons could be different from platform to platform.

The source code is just **an implementation** of the specification, and this already makes it difficult to discern between platform constraints and protocols design. Moreover, in the actual coding, there are usually **many layers** overlapping for historical reasons, such as patches and fixes, inelegant “hacks” to temporarily solve peculiar problems of a particular application which have no general meaning, misaddressed backward compatibility needs, bugs, meaningless code (there are tools to assess what part of a software program are seldom or never invoked during execution by a debugger) and so on. Lastly, without good guidance, it is almost impossible, at least a painful experience, to find out what particular library or object does perform a given task and on which condition, in a codebase which can easily run in the thousands of files an in the millions of programming lines.

Because the source code is only *an* implementation of the specification, sometimes it is a *bad* implementation of certain specific points. This is not uncommon, on the contrary, it is quite customary because the release process of software is quicker than the process of finding bugs and errors. For solving this problem, usually companies, including Microsoft, publish sometimes large “fixpacks” or “service packs”, sometimes small fixes. This increases the entropy of source code also timewise, so that it is even harder to keep track of the “current version”, because one needs to firstly assess if the later fix has resolved a design error, a bug, a need to redesign some part of the code because the requirements of a specific application. We do not see any commitment in the proposed agreement dealing with the time of delivery of subsequent changes and with the illustration of what is the purpose of such change. In addition, the impossibility to print, store or otherwise keep track of the inspected material is a serious impairment in keeping evidence of non-compliance.

In further addition to the point above, the **impossibility to actually compile** the “licensed” code means that there is no indication that the code is genuine or ways to ensure it. For as clever and experienced as a programmer can be, he/she cannot tell if the provided source code is authentic or forged, unless this is passed to a compiler, it compiled successfully and it works flawlessly once the dependencies are met. Therefore this unlucky programmer would try to infer possibly wrongful information from “scam”

source code, he/she has no particular reference for judging the quality of the gathered information, he/she makes a very good job of producing some implementation which fails to interoperate with the original product. Who can say, honestly, if he/she has misjudged the source code or if the source code was a big fat fake?

In our opinion Microsoft **has chosen to impose its own standard as a *de-facto* standard**, it has **the burden** to make sure that this standard is **provided on a fair basis to externals as it is to internals**, without any distinction. It has decided to make a poor job of a standardization body, it has decided not to work upon specifications but to implement some rough ideas without properly documenting all the details of it, **it cannot complain** that it is not in a position to do the reverse process. If it is impossible to it to give sufficient information of what it has put on and imposed to the market, and which cannot be now avoided, it is its obligation to remedy this situation. The standard upon which the compliance shall be assessed are therefore **the results** that the compliance brings in terms of allowing competent technical experts to successfully implement interoperable solutions. In other words, **an obligation of result**, or, otherwise, a reverse burden of proof that the inability to produce a successful implementation of the standard is due to ineptitudes of the competitor and not to poor information provided.

Microsoft has not sought guidance on how to design its protocols, it has decided to do it at its please and to satisfy only its needs, while **seeking dominance** (also) by technical tie-ins (which is already a byproduct of lack of interoperability) and without caring the least of interoperability, or, worse, avoiding interoperability as much as possibly. It has willfully excluded anybody from this process, now it must bear the consequences and t**ake the entire blame**. It cannot hide behind the impossibilities it has caused (according to the principle of “*sibi imputet*”).

Question #2

Please indicate whether you consider the terms of the source code licence reasonable.

We find that the legal conditions are still unreasonable. The reference license conditions *per se* could be free of criticisms from the standpoint of FSFE and the Samba Team. As we have always said, we have no desire at all to see Microsoft's source code, but respect the decision of others who would want to do it. As we suppose that these others would try to implement an alternative for other operating systems / platform, the use of the source code as a reference could seem reasonable in general terms.

However, because the software design has certain constraints which cannot be avoided if the competing solution has to perform certain activities to interoperate, the impossibility to use the same solution because of the legal conditions could lead to copyright infringement claims or to the impossibility of implementing a competing and adequately efficient solution. The reference made by the agreement when allowing to use identical solutions that “are mandated by functional constraints of the WSPP Protocols” leaves large room to doubt that functional constraint not directly imposed by the WSPP Protocols, but by other reasons, yet unavoidable, could not give adequate reason to use identical or strictly equivalent source code, *i.e.*, falling under copyright infringement.

We see completely unreasonable also the clause that forbids:

> *“ (b) to intentionally use its access to Source Code under this Source Code License to determine whether any Microsoft product containing the Source Code infringes any claim of a patent or patent application, or use information obtained from such access to assert patent infringement claims against Microsoft;”*

While FSFE is **strongly against software patents**, various Free Software companies hold a valuable portfolio of computer-related inventions, which can find in parts of the software indications of an infringement. The language of the clause, although it uses the “intentionally” specification, is vague enough to prohibit any use of the information for evidence gathering of any sort, and resembles too closely to a “**not to challenge**” clause. The effect being the unavoidable option to waive any patent claims against Microsoft or to risk advanced termination of the agreement.

Further, this clause seems to have no meaning:

> *“ii. Patent License: only in addition to a patent license which Licensee has obtained under the Agreement, [Microsoft grants] the right, under any claims of a Microsoft patent or patent application that cover functionality in that Source Code used by Microsoft in implementing the WSPP Protocols for purposes of Windows interoperability, to use information in that Source Code solely to make, use, sell, offer for sale, import or otherwise dispose of or promote LSIs”.*

The right to make, use, sell, offer for sale, etc. a LSI implementing a patented invention is ostensibly already granted by the Patent license, which is an option of the WSPP Licensing scheme. Revealed source code, by definition, **cannot add anything relevant to a valid patent**, because either sufficient information to use the invention covered by the patent is already made public by the patent application and/or by the technical description, or the patent is invalid. If a licensee must access the secret source code to implement the invention, or in other words it is enabled (also) by the source code reference license (which is a trade secret license), it means that a substantial part of the invention is not properly disclosed, **thus the patent is null and void**. Requesting a license for a knowingly void right is a blatant antitrust infringement under any possible meaning. Thus we suspect that either this clause is useless, or that the claims of Microsoft's to be protected by patent law against the Decision are not worth the paper on which they are written.

The two clauses, read together, mean that the licensee, for obtaining nothing more than what should definitely be already granted by the ominous patent license, waives the right to challenge either the patent and the patent license. If the licensee challenges the patent or the patent license, it *could* have gathered the grounds for said action by accessing the secret proof of violation, while secrecy *is* the violation. That is clever, but *unreasonable*.

Apart the above specific criticism, the conditions inherit the same flaws of the entire WSPP because **they incorporate by reference** the same conditions of the main WSPP license scheme, which is designed to exclude Samba from access. There is no point therefore to discuss these conditions without a 180 degrees turn in the main scheme.