

Sovereign Software

Open Standards, Free Software, and the Internet

Georg C.F. Greve <greve@fsfeurope.org>
Free Software Foundation Europe, President

1 Introduction

Software issues are issues of power and fundamentally shape the societies we are living in. Even to those who had not followed digital policy issues before this became increasingly evident throughout the United Nations World Summit on the Information Society (WSIS). Two fundamental questions characterise this battlefield: Who controls your data? Who controls your computer?

The first question generally revolves around Open Standards, and in particular how they should be defined and upheld. All players in the field speak out in favor of Open Standards, but some wish that term to be understood in ways that they still control your data and retain the power to lock out competitors at will.

The second question has been one of the key controversies throughout the WSIS, it was highly controversial during the WGIG, and remains controversial throughout the Internet Governance Forum (IGF). This issue is one of software models, of proprietary vs Free Software, and has been oddly polarised between for-profit and non-profit in the WSIS context.¹

Some people see the two issues connected, other argue they should always be treated separately. As will become clear later on, the two issues are indeed not connected in theory, but have a connection in practice. In order to understand this, it is important to consider them isolated and individually first.

2 Open Standards

Having been preached as commonplace statement in the information technology industry for many years already, Open Standards only recently made their entry into the center stage of public policy. One of the places where this happened was during the WSIS, and will be of major importance for the Internet Governance Forum (IGF). But why are Open Standards so important?

2.1 Background on formats

All computers store and transmit information in encoded form. These used to be very simple representations where certain numerical values stand for a certain character, for instance. And while their complexity has been increasing steadily with the power and complexity of computers, certain basic rules always apply.

¹This may have been due to the specific situation that mainly the largest proprietary software multinational followed the WSIS intensively while the large multinational vendors of Free Software generally did not participate and were thus not represented in the CCBI.

The first important rule is that any such choice of encoding is an arbitrary, and not a natural choice. The number 33 may represent the letter 'a' or 'z' depending on the convention for this standard. There is no right way of doing this, there are only possible ways.

The second important rule is that once data has been encoded in a certain format, it can only be read by software that implements this format, and implements it exactly. Even slight deviations from the conventions of the format will easily cause massive data corruption. A common and mostly harmless form of this is lost or broken formatting in text processing software. In the worst case the data will be unrecoverable.

2.2 Formats and market failure

From a market point of view, such a situation generally brings about market failure: Customers who saved their data in one format quickly find themselves unable to choose another vendor that was not able to implement the same format, or unable to implement it well enough. If the only way to migrate is to lose years of data there is a very effective vendor lock-in that practically makes it impossible to choose software according to its merits.

Additionally, strong network effects dominate today's computer world. If a company invested heavily into a desktop infrastructure in the past and this infrastructure uses certain communication protocols, they find themselves faces with two alternatives: Get only such software that implements these protocols perfectly or write off the investment and replace the entire infrastructure, obviously at a high additional investment.

A third party vendor that wishes to enter this market is faced with a situation similar to someone finding themselves in a room of people speaking a foreign language, with no dictionary and syntactic help available. Human languages are collections of arbitrary decisions just like computer formats and protocols. There is no inherent natural reason to call a table a table, or call a chair a chair. For someone not speaking that language and without a dictionary or at least someone willing to explain the language it becomes very hard to communicate.

In information technology, some people have been able to divine information about such protocols and file formats merely by watching others use that language. This is called protocol analysis and has helped mitigate the negative influences of the systematics above somewhat.² It is also the reason why some dominant vendors start inserting cryptography into their protocols, preventing further protocol analysis in the future.

2.3 Public Policy implications

All of this is obviously a major concern for public policy for various reasons and has been discussed in various fora, e.g. the Danish parliament for its motion *B 103*³ in which the following reasons are elaborated.

²This is how OpenOffice (<http://www.openoffice.org>) came to its ability to generally read most documents written with Microsoft Word, for instance, or how the Samba (<http://www.samba.org>) software became able to replace large parts of the functionality of Microsoft workgroup servers.

³<http://www.ft.dk/Samling/20051/beslutningsforslag/B103/index.htm>

2.3.1 Healthy procurement policy

It is obviously not sustainable to make investments that will become subject to the effects explained above. There is virtually no market and a single vendor is in the position to vitriate the entire investment. As this is not in line with the principles of efficient and sustainable procurement by the public sector, such situations have to be avoided.

2.3.2 Protect democracy from networking effects

The same networking effects that were described above take place when the software needs to communicate with citizens. Only citizens that choose the one vendor implementing that proprietary protocol would then be able to communicate with their administration, violating the basic principle of citizens being able to freely communicate with their governments. Using proprietary formats and protocols would instead force them into the same vicious cycle of investment and increasing stakes explained above.

2.3.3 Ensure open competition

Such a situation is obviously contrary to the principles of open competition and markets and will quickly bring issues of market concentration and stifling of innovation. As this is contrary to the goals of any government, governmental procurement should support open and competitive markets.

2.3.4 Merging effects, ensuring accessibility

In the scope of more efficient administration, many municipalities and different parts of administration are starting to pool resources. If this is attempted with proprietary formats, it usually means that unless all parts have already been using the same software, significant investments by one or several of the administrations would be lost.

Also all of this will have to take into account the rights of people with disabilities, who may have special requirements in software that the implementation of that proprietary format may not meet. In this situation there will be no possibility for people with disabilities to communicate with their governmental services.

2.3.5 Commercial-political perspectives

Ultimately there are strong political issues with storage of data in proprietary formats. What if those data become inaccessible in the future due to problems with that particular vendor? Can a government really rely blindly and without alternative on the goodwill of any singular commercial entity?

2.3.6 Long term commercial aspects

Also, with all of the above, increasing choice and freedom to choose in an open market will bring additional long-term commercial benefits.

2.4 What is an Open Standard?

There are various definitions for what should or should not be considered an Open Standard. The aforementioned Danish motion describes it as:

- well documented with its full specification publically available
- freely implementable without economically, politically or legal limitations on implementation and use, and
- standardized and maintained in an open forum (a so-called standards organisation) through an open process.

This is relatively similar to the definition of an Open Standard by the European Commission in its European Interoperability Framework.⁴

Both these definitions were criticised by the vendors that profit commercially from the dependency cycles explained above, as well as organisations representing their interest. The usual argumentation for this criticism is generally oriented along the lines of patents that were granted on such a format or protocol, and for which the patent holder might choose to generate license revenue. The euphemism du jour for this is usually "Reasonable and Non-Discriminatory" (RAND) licensing.

This is but a euphemism because patents are by their nature limited monopolies granted by law to a single entity. This entity will always have the upper hand in any dispute, and indeed there are plenty of stories about formats and protocols that are theoretically known, but remain proprietary due to patent issues.

That all other vendors not holding this patent are put in an equally bad position may indeed seem non-discriminatory, but it does not fundamentally change the balance of power of the situation.

All formats and protocols are fundamentally arbitrary in nature, but must be followed precisely for the data that was stored in them to be recovered.

2.5 Open Standards in practice

In theory, the definitions of the European Union or the Danish parliament would be sufficient to define an Open Standard. In practice things have proven to be more complicated because the situation with proprietary formats described above is immensely profitable for the vendor in control of that software.

So ultimately, a proprietary vendor with a certain amount of market penetration has an economic incentive to violate the Open Standard and turn it into a de-facto proprietary one. This indeed has happened repeatedly in history. The European Commission antitrust investigation against Microsoft provides testimony to how deviating from an Open Standard (CIFS, the "Common Internet File System") allowed Microsoft to leverage its desktop monopoly into near total dominance on the workgroup server market. This has proven so profitable that Microsoft appears more inclined to pay billions in fines than to stop this practice.⁵

⁴<http://ec.europa.eu/idabc/en/document/3473#finalEIF>

⁵<http://fsfeurope.org/projects/ms-vs-eu/>

Often this is also done by slightly changing the implementation in ways that are hard to pinpoint or can be debated within the limits of human interpretation, but make sure that the implementations of other vendors will not integrate flawlessly anymore. The economic incentive for this is huge for proprietary players that bypass a certain threshold in size.

2.6 How to maintain an Open Standard

The only way to prevent this sort of thing seems to add one more criterion to the definitions above: "The standard must have at least one Free Software implementation and all implementations that seek to be compliant with the Open Standard must be regularly tested against the Free Software implementation(s), which act as the common reference base."

Because Free Software⁶ is, inter alia, defined by the freedom to study its implementation, this allows all players in the market to study the common reference base not only in specification language, but also in language, and regular tests against that base can help curb deviations from the Open Standard.

Free Software also provides the freedoms of use, modification and distribution, therefore most vendors can also simply include that implementation in their own software, further reducing interoperability barriers.

So while there is in theory no connection between Open Standards and Free Software, in practice Free Software becomes a necessary component to maintain an Open Standards against economic incentive to proprietise or deviate from an Open Standard.

2.7 Open Standards and the WSIS/IGF

A good example for this is the internet. Before the internet became what it is today there were various different attempts to establish something similar. Why did the internet succeed? Because the implementations of basic internet protocols such as TCP/IP were Free Software and therefore equally available to all.

The World Wide Web repeated this story when Tim Berners-Lee waived all patents on the protocols and formats, and they were implemented in Free Software. More than 60% of the world's web sites run on Apache, one of several Free Software web servers.

Sadly enough, the language on Open Standards adopted in the WSIS and subsequently carrying into the IGF would not be sufficient to build something like the internet. Formats and protocols going by that definition would be subject to all the effects elaborated above.

So it is important that the Internet Governance Forum (IGF) now goes beyond this insufficient language and works out true international consensus that will protect the internet from "propertisation creep" in all its protocols and formats. Open Standards are an essential building block of the internet – they must be maintained for the internet to not fall victim to a tower of babel syndrom.

⁶For a full and concise definition of Free Software please consult the "Free Software Essentials Reference" also supplied in the substantial contributions to the IGF.

3 Free Software

The practical connection between Free Software and Open Standards has already been elaborated, but there are other, genuine Free Software issues that have no direct connection with Open Standards. These are issues of software model and ultimately of control over your own computer.

Free Software is software that gives all users and developers the following four freedoms:

1. **The freedom to run the program, for any purpose.**
2. **The freedom to study how the program works, and adapt it to your needs.**
3. **The freedom to make and redistribute copies.**
4. **The freedom to improve the program, and release improvements.**

It is important to note that any of these activities can be commercial, indeed there are large international companies for which Free Software is a very profitable business, IBM, SUN, HP and others among them.⁷

3.1 The difference of software models

So commerciality is not the dividing line between proprietary and Free Software. In the ultimate abstraction the issue of software models comes down to one fundamental question: Who has control over the software that runs your computer?

With proprietary software, that is always and exclusively the proprietor of the software. The owner of the computer generally gets some usage permissions for certain purposes, but these can usually be revoked and the user never owns or controls the software in any meaningful sense. With Free Software, the user is put in charge and control of their own software.

This shift in power from "one over everyone else" to "everyone over themselves" fundamentally affects how national economy, enterprises, science, education, politics and society as a whole works. A full elaboration of these issues would be beyond the scope of this paper, so it will focus on a few selected issues of governance and sovereignty.

3.2 An issue of control

Although this may seem like an obviously falsehood, there is widespread common belief that the user controls their computer. In reality, it is only the software that actually controls the computer, taking some hints from the user if so programmed. This is an important fundamental distinction, because it makes clear that only by controlling the software can users control what their computer actually does.

There are plenty of examples of software doing things secretly, and without the knowledge of the user. One recent example includes a piece of software that comes with SONY CDs and informs SONY every time that CD is played, and on which machine. All of this happened without visible signs on the computer, and without any information for or agreement by the

⁷A more complete and elaborate definition of Free Software and a clarification of the most common misunderstandings is available on the "Free Software Essentials Reference" sheet also in the substantial contributions to the IGF.

user. Indeed, the user was falsely informed by SONY that this did **not** happen until someone was able to prove them wrong.⁸

Similar stories exist for various other proprietary software solutions, including collaboration and conferencing software that was allegedly safe and highly encrypted and most likely used by governments for confidential activities around the world.

Because there is no way to know for sure what your software does unless you have full control over it, the German Agency for Security in Information Technology (BSI) has a recommendation for Free Software.⁹ Indeed, the German embassies around the world are networked with the German government through Free Software, using the GNU/Linux based SINA box.¹⁰

3.3 Issues of political mandate

Even though there has been considerable movement on the issue, Open Standards in public administration are still the rare exception. And in the proprietary world, which is still the norm in many governments, generally only one vendor can provide software that will be able to access those data and processes. So effectively much of public administration and governmental processes are controlled by software which in turn is controlled by only one vendor that the government has no meaningful control over.

Free Software is the only way to ensure that governments actually control their own data and processes, including critical infrastructures. Free Software also avoids the aforementioned "proptertisation creep" on Open Standards: There is no profit in this, as generally any vendor can choose to supply or maintain that solution.

Only Free Software is ever truly Sovereign Software.

3.3.1 Free Software and the WSIS/WGIG/IGF

Free Software and the internet go hand in hand. It was Free Software that critical to making the internet possible, and indeed Free Software continues to shape and run the internet. At the same time, Free Software and its representatives has been all but excluded from the WGIG and the IGF processes thus far.

If the Internet Governance Forum is to become a truly inclusive forum to discuss internet related issues, Free Software and its representatives should be included in all relevant fora and all political levels of the IGF. Otherwise there is a possibility that the people who actually continue to build the internet will simply take their discussions elsewhere.

⁸<http://www.wired.com/news/privacy/0,1848,69601,00.html>

⁹<http://www.bsi.bund.de/oss/index.htm>

¹⁰<http://www.bsi.bund.de/fachthem/sina/index.htm>