European Telecommunications Standards Institute

Open Source impacts on ICT standardization

Final Report – Analysis Part - VA6

December 12, 2005

Hervé Rannou
Jean-Baptiste Soufron
ITEMS International
46 ave Pierre Brossolette
92240 MALAKOFF
Tel: +33 (0)1 4985 0326
Fax: +33 (0)1 4985 1407

hrannou@items-int.com
www.items.fr

SOMMAIRE

This report is the result of the study carried out by ITEMS International and mandated by ETSI. In any case, it does not represent the position of ETSI on the issues described or analyzed by ITEMS International.

ACKNOWLEDGMENTS

This study has been carried out in collaboration with with Margot Dor, Director Business Development and Stéphane Tronchon, Legal Counsel in ETSI

We would like to thank the following people for their inputs[1] and valuable comments:

ETSI
- Dr. Karl-Heinz Rosenbrock, Director General,
- Yves Chauvel, Director for European co-ordination,
- Gavin Craik, EC/EFTA Mandates Manager
- Jørgen Friis. Deputy Director-General
- Patrick Guillemin, ETSI Plugtests Coordinator
- Gerd Ochel, European Co-ordination Officer
- Adrian Scrase, Chief Technology Officer, 3GPP Secretary
- Anthony Wiles and Ultan Mulligan, Protocol & Testing Competence Center

BSA
- Benoit Muller, Brussels

CompTIA
- Hugo Lueders, Director European Public Policy, Brussels

Computer & Communications Industry Association
- Ed Black, President, Washington DC
- Will Rodger, Director Public Policy, Washington DC

European Commission
- Paul Timmers, DG Information Society, Head of Unit for eGovernment
- Bernhard Schnittger, DG Enterprise and Industry, Head of Unit for IDABC

Free Software Foundation Europe
- Ciaran O'Riordan

French Telecommunication Regulation Authority (ART)
- Didier Chauveau, Head of prospective unit, Paris

IBM
- Karl Cox, European Governmental Programme, Paris
- Michael Breidthardt, Stuttgart

INRIA
- Bernard Lang, Directeur de Recherche, Paris

Japanese Ministry of Economy, Trade and Industry (METI)
- Takashi Kume, Deputy Director, Tokyo

Institute of Sofware Chinese Academy
- Dr Zhao Chen

Microsoft
- Malcolm Bermange
- Steve Mutkovski, Corporate Attorney, Intellectual Property & Licensing Group

Nokia
- Kari Lang, International Standardization Relations

Sun Microsystems
- Verina Horsnell

University of Limerick
- Brian Fitzgerald, Chair in Innovation in global Business & Technology

US Administration
- Jeffery Goldthorp, FCC, Chief Network Technology Division, Washington DC
- Christina Speck, NTIA Office of International Affairs, Washington DC

Vodafone
- Alan Cox, Head of International Standards

W3C
- Rigo Wenning, Legal Counsel, Sophia-Antipolis

ZOPE Europe Association
- Paul Everitt, Louvain

[1] Interviews or documents transmitted

1 Introduction

ICT is a fast growing industry involving global and local players from the IT, Telecommunications and Multimedia sectors. The convergence factor is inducing changes in the business processes and models that impact strategies and methods of industrial players and standards-makers as well.

One of the key elements in convergence is the ***growing role of software***, in particular in the telecommunication sector and in consumer electronics. Today, interfaces and components developments are mainly software-based. R&D laboratories need more and more software engineers, sometimes creating a cultural and generational gap within ICT entities.

Another major change is occurring in the software sector with the advent of Open Source software. Originally initiated by Internet R&D communities, Open Source now seems to pervade the whole ICT market and does benefit from the support of a series of major industrial/commercial players.

In the same way, Internet has not encompassed the whole ICT sector but has changed dramatically its economics and its models. Open Source does not include the whole software market, but is a different approach that is radically changing the rules of the game.

The impact on standardization could be significant in the near future. The border between Open Standards and Open Source is becoming the field of new challenges for both private and public players. New rules are emerging that will impact the definitions of Open Standards in the future.

The objectives of this study are to analyze the underlying issues in this debate, including:

- the background of this debate
- a typology of Open Source Licenses
- the IPR issues
- the impact on standardization rules

2 Open Source Software

2.1 DEFINITIONS

First of all, it must be underlined that Open Source doesn't mean free to use. Actually, Open Source is a complex world with many kinds of licenses that can be free of charge but that always include complex provisions for both parties.

Generally speaking, a license allows a holder of Intellectual Property on a technology to grant a second party the right to use it. The license usually specifies the conditions under which the technology can be used or disseminated to third parties, payments to the licensor, whether modifications are allowed, the risk and liability each party assumes, representations and warranties, as well as provisions for support and maintenance.

The various kinds of Open Source Software licenses are not all interchangeable and they must be clearly identified:

Freeware involves no payment and allows the user to replicate and distribute the software freely.

Shareware is provided free, but a voluntary payment is requested if the user puts the software to productive use.

A source code license grants access to source code for the licensee. ***Open Source licenses*** can be defined as a specific set of terms and conditions that meet the requirements summarized by the Open Source Initiative[2] that were first set by Bruce Perens in June 1997[3].

Similarly, ***Free Software licenses*** were defined by the Free Software Foundation and they essentially include four requirements[4]:

1. the right to ***access the source code*** - the right to get the source code and to read it. This right is the main component of any Open Source software since it is a necessary preliminary step to all of them. All Open Source licenses grant access to the source code of the software.
2. the right to ***modify*** – the right to change the source code in order to correct it or to adapt it. Most Open Source licenses recognize this right

[2] http://www.opensource.org/
[3] http://www.opensource.org/docs/
[4] http://www.gnu.org/philosophy/license-list.html

but not all of them. For example the Perl Artistic license states that if a package is modified in a way that it changes from a "Standard Version", then these changes must be posted in a very specific and detailed way described in the license.

3. the right to ***redistribute*** – the right to copy and distribute the software. Licensors often modify this right in many ways. For example, the Sun Community License allows licensees to distribute software only as long as they make no commercial gain directly from it.
4. the right to ***use*** – the right to compile the source code and execute it. Licensors can adapt this clause as per the redistribution clause.

Each Open Source license is a different combination of these four elements according to specific legal provisions and industrial objectives.

The primary issues in choosing license terms and conditions are whether the licensor wants revenues, whether it allows modifications to the source code and whether it allows the source code to be "forked" in different versions.

Free Software can be defined as software distributed with a license that will implement these four rights. For example, the IBM Public License[5] and the Mozilla Public License[6] are Free Software licenses.

Copyleft is a concept drawn by Richard Stallman and the Free Software Foundation[7] as a license that reiterates the four freedoms described above and that must be reproduced in identical terms in the case of redistribution. This avoids the distribution of the modified software restricting the initial rights***. Copyleft Software*** can then be defined as software distributed with a license that will mandate licensees to publish derivative works under the same terms. For example, the GPL[8] and the LGPL[9] Licenses are Copyleft Software Licenses. The Free Software Foundation is the strongest advocate of Copyleft and assumes intellectual leadership over this movement.

Copyleft licenses essentially preclude the ability to draw revenues from the code itself but admit other business models. ***Community Licenses*** and ***Examination Licenses*** are interesting for paid software licenses as a way of garnering customer or community assistance in improving software and encouraging complementary research activity.

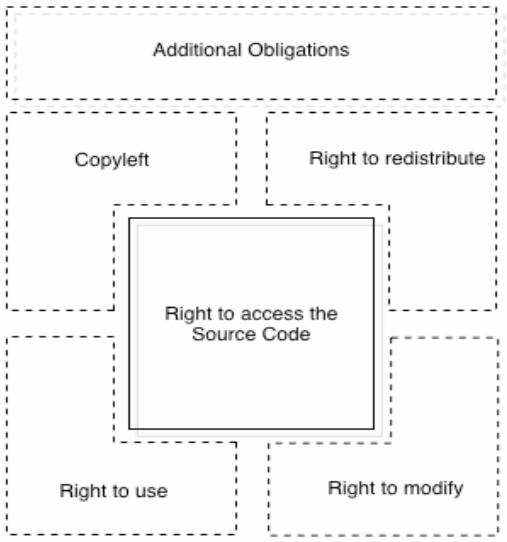
---

[5] http://www.opensource.org/licenses/ibmpl.php
[6] http://www.mozilla.org/MPL/MPL-1.1.html
[7] http://www.gnu.org/copyleft/copyleft.html
[8] The GNU General Public License (GNU GPL or GPL) is a free software license, originally written by Richard Stallman for the GNU project (a project to create a complete free software operating system). Released in 1991, the version 2 became the most popular licenses for free software.
[9] Released in 1991 and updated in 1999, the GNU Lesser General Public License (LGPL) is a commonly used software license. It is originated from a modified version of the GPL in order to apply to software libraries.

At the end of the day, most licenses are a complex mix of Open Source, Copyleft and Free Software.

Furthermore, most licenses also allow for a variety of **additional obligations**. For example, the Creative Commons NC License only authorizes non-commercial uses of the work.

Finally, an Open Source software can be represented as follows. The right to access its Source Code could be seen as its central core and all other rights granted by the licensor are optional and can be modified. The licensor can also add some other obligations that will add up to the classical rights defining Open Source and Free Software.



It must be noted that the Open Source Initiative (OSI) provides a more detailed framework describing an Open Source License:

1. **Free redistribution:** the license shall not restrict any party from selling or giving away the software as a component and it shall not require a royalty or other fee for the sale of the Software itself.
2. **Source Code:** the program must include source code, and must allow distribution in source code as well as in compiled form.
3. **Derivative works:** the license must allow modifications and derivative works.
4. **Integrity of the author's Source Code:** the license must explicitly permit distribution of software built from modified source code. The License may require derived works to carry a different name or version number from the original software.
5. **No discrimination against persons or groups:** the license must not discriminate against any person or group of persons.

6. **No discrimination against fields:** the license must not restrict anyone from making use of the program in a specific field.
7. **Distribution of license:** the rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.
8. **License must not be specific to a product**
9. **The license must not restrict other software:** for example, the license must not insist that all other programs distributed on the same medium must be open source software.
10. **The license must be technology-neutral:** no provision of the license may be predicated on any individual technology or style of interface.

The differences of terminology between OSI and Free Software Foundation do not imply any practical differences *per se*. Free Software programs and Open Source Software programs refer to the same reality.

Supporters of the “Free Software” term use it to emphasize its four freedoms as ethical justifications whilst supporters of the “Open Source” term emphasize its operational characteristics without attempting to supply any ethical justification.

Given that the OSI provides a list of certified licenses and that the most important thing is to understand the various provisions enforced by a given License, it seems to be a more interesting approach to use the simple FSF classification and to state when a License is approved by the OSI.

It must be noted that even if the FSF played a prominent role in this debate, OSI became more influential over time, in particular since January 2005 when IBM decided to allow royalty-free use of patents in any software covered by the Open Source Initiative's Open Source Definition[10]. A complete list is given in Annex.

[10] http://www.ibm.com/ibm/licensing/Patents/pledgedPatents.pdf

2.2 SUN, MICROSOFT, BSD, GPL AND OTHERS

A number of sub-models of open source licences can be found on the market. For example, the ***Sun Community Source License***[11] was put forward by Sun to promote its Java Technology. It follows a Community model specifying three level of participation:

- research use (examination and evaluation of the source code)
- internal deployment (right to manipulate the code during development and testing)
- commercial use (use of the code in a commercial product triggering royalties back to Sun)

Under the SCSL, licensees must contribute maintenance improvements to the community but they are free to make proprietary improvements or extensions as long as they use open interfaces. Sun's motivation is to gain wide participation in advancing and promoting the platform whilst maintaining a degree of control and reaching for revenues.

Similarly ***Microsoft offers a Shared Source License*** for .NET Common Language Infrastructure and its associated C# compiler[12].

Neither the Sun SCSL[13] nor the Microsoft SSL[14] are OSI certified Open Source Licenses.

As a matter of fact, the SCSL does not qualify as OSI OSS or as Free Software because contributors are required to maintain compatibility with the Java standard, **as defined by Sun**.

On the contrary, ***Sun Public License*** is an OSI OSL and ***Sun Industry Standards Source License*** both qualify as FSF and OSI OSL.
On the other hand ***BSD***[15] or ***GPL*** are both FSF and OSI OSL when neither really ensures compatibility.

---

[11] http://www.sun.com/software/communitysource/principles.html

[12] Microsoft's Shared Source provides various source code licensing options ranging from restrictive licenses designed to enable the viewing of source code without modification all the way through full commercial derivative rights terms. They recently standardized their source code licenses to three template forms: Permissive, Community, and Reference. None of them have been submitted to the OSI at this time.
http://www.microsoft.com/sharedsource.

[13] http://www.oreillynet.com/pub/a/oreilly/java/news/loukides_0399.html

[14] http://www.opensource.org/docs/sharedsource.php

[15] Updated in 1999, the BSD license is used to distribute BSD software. The owner of the original BSD distribution was the "Regents of the University of California" because BSD originally came from the University of California, Berkeley. The license is very permissive, it authorizes commercial uses, it does not require any copyleft provision and derivative works

For example, Linux was never certified by the Open Consortium for compliance with POSIX standards[16]. The point is that if GPL permits compatibility with other standards, it does not specifically encourage it. Actually, due to its fight for orthodoxy, the Free Software Foundation contributed to the proliferation of incompatible programs with a very similar functionality by promoting forks of important free/open products.

In fact, rather than adherence to standards, the founder of FSF, Richard Stallman, successfully used the "embrace and extend tactic" in GCC[17] and other products. FSF proponents would not try to follow standards nor take steps to ensure protection of their softwares from deviation.

Yet for some open source promoters, it made sense to add some protection against deviation from the standard into the license by restricting the rights for the creator of derivative works. As discussed, in addition to promoting Java, the idea of the previous mentioned ***Sun Community Source License*** was to control its developments. If one can take a program, modify it and then choose how to release the modifications, one can make them available under the same license but only in object code form. Sun, however, makes it mandatory to distribute it including the 'standard version' with copyrights and disclaimers.

For Sun, the next step to ensure standard compatibility was made with the ***Sun Industry Standards Source License*** (SISSL) that was created in early 2001. The license basically says that an implementation should conform to an approved industry standard, or if it doesn't, then the developer must document all deviations from the standards and provide a source code of the reference implementation.

As clearly shown, all those licenses grant different provisions and they can be incompatible.
Thus, when a problem of compatibility between different licensing models occurs, a work is sometimes published using ***Dual Licensing***.

Such is the case for the Mozilla Public License (MPL) that authorizes the initial developer to designate parts of the original or modified code as covered by a license of his/her choice (including a proprietary license). Because of this partial appropriation of certain developments, the MPL has not been designated as GPL compatible, and the Mozilla community publishes developments under Dual Licensing with three Licenses: MPL,

---

can even appropriated. This license is very common and there are many examples of BSD code in Microsoft products (like parts of the networking code) or within Mac OS X (using parts of FreeBSD).

[16] POSIX: IEEE initiative to standardize Unix. http://posixcertified.ieee.org/

[17] GNU C Compiler

LGPL and GPL. The SISSL is also often quoted as being an ideal choice for Dual Licensing and it is used with the LGPL in the OpenOffice.org project.

2.3 Open Source Business Models

Open Source Business Models can be described as follows:

- Service Model: The software is released for free or at low cost, but it can be used to sell associated services. The originating company of the software will profit from its knowledge of the software to provide better services than others. This is a common model used with the GPL or BSD License and it is used with success by companies like RedHat.
- Integration Model: The software is released for free or at low cost but its license will include a copyleft clause allowing a third-party to integrate the software in a different proprietary product. This model is not common but it features a few famous examples like Apache and Websphere.
- Product Model: The software is sold as a product. This model is not common and essentially adopted by players who already have a dominant position on the market such as Apple with OSX or Sun with Java.

	Open Source Strategy	Business strategy
Service	- To develop service using **GPL** or other **existing OSS component** - To develop **own "OSS" architecture**	- To **promote openness** - To develop **ad-hoc architecture and software to costumers** - To **reinforce the critical relationship** with the costumers
Product	- To develop **specific component** integrated in a **proprietary product** - To **promote** theses components within the **OSS communities**	- To **strengthen the position of the product** on the market - To benefit of **OSS communities assessment**
Integration	- To develop **Open Source Enterprise architecture** - To **develop service** using their own architecture mixed with others (OSS or not) - To **promote their own architecture** including components within the OSS communities	- To be perceived as a **federator** - To **strengthen their position on existing customer base** - To take an **overall commitment** on the evolution of enterprise IT architecture

ITEMS International - 2005

A majority of players and analysts claim that the Service Model is the only efficient model but a deeper analysis can help establish relationships between Open Source Business Models and Open Source Licenses.

2.4 TYPOLOGY OF OPEN SOURCE LICENSES

As previously discussed, Open Source licenses follow simple rules, but their combination leads to complex situations.

The main criteria for summarizing OSS licenses and classifying them can be presented as follows within the FSF framework:

Licence	Free Software				FSF Copyleft	OSI Certified	Economic Model
	OSS	OSS Flavours					
	Access	Use	Distrib.	Modif.			
BSD 99	X	X	X	X	-	-	Service
LGPL	X	X	X	X	X	X	Integration
GPL	X	-	X	X	X	X	Service
W3C SL	X	X	X	X	X	X	Service
BSD	X	X	X	X	-	X	Service
Apache 2	X	X	X	Specific	X	X	Integration
IBM Public License	X	X	X	Specific	-	#	Integration
Sun Industry SSL	X	X	X	X	X	X	Service
Sun Public License	X	-	X	X	-	X	Integration
Sun Community L	X	X	-	-	-	-	Product
Apple Public SL2	X	X	X	X	X	X	Product
Microsoft SSL	X	-	-	X	-	-	Product
Mozilla Public License	X	Specific	X	X	Specific	X	Service
ITEMS International - 2005							

2.5 TYPOLOGY OF SDO IPR

Mirroring the typology of Open Source Licenses, it can be interesting to draw a typology of the current situations of some Standard Setting Organizations with regards to their IPR policy.

The main questions to answer are to know what kind of Intellectual Property is addressed by their IPR Policy (Patents, Trademarks, Copyright, etc.), if they can include Intellectual Property in their standards and what are the licensing provisions required to participate in a standard.

SSO	Policy	Can Standard include IP?	Licensing Provisions
ANSI	Patents	Only for technical reasons	RAND, ANSI will review claims of unreasonableness
CEN	Patents	Exceptional	RAND or withdrawal of standard
ETSI	Patents, Copyright	Yes	RAND irrevocable
I2O SIG	Patents, Trademarks	Yes	Royalty Free
IEEE	Patents, Copyrights	Yes	RAND, terms must be specified
IETF	Patents, Copyrights	Yes	RAND, terms must be specified
ISO	Patents, Trademarks, Copyrights	Yes for Patents, no for TM	RAND Patents, non-exclusive Copyright, no TM provisions
ITU	Patents	Yes	RAND and no-monopolistic abuse
RosettaNet	Patents, Copyrights	No	Patents assigned to RosettaNet
W3C	Patents, Trademarks, Copyright	Yes	Royalty Free

RAND: Reasonable and non-discriminatory policy
TM: TradeMark, C: Copyright
ITEMS International

It is clear that these SSOs work on similar topics (Patents, Copyright and Trademarks), but they have very different policies even when they are similar in some ways (each RAND system is a different one).

3 Copyright Open Source and IPR

3.1 Open Source is Copyright IPR

It is often said that Open Source is not compatible with IPR Policy. As previously mentioned, free software does not mean free to use, it means software distributed with a license authorizing several uses in advance. Open Source is based on copyright rather than on patents but ***it is also subject to IPR.***

ETSI already recognizes some role to copyright in the articles 9[18] and 11[19] of its IPR Policy, even if in ETSI case, the emphasis is rather on patents for the time being.

As shown in the typology above, this conflict between patents and copyright shows both the technical and legal aspects of the use of Open Source Software. This is due to the fact that with some of the early licensing models it was very difficult to protect the company know-how and IPR.

Today, there are many different kinds of licenses to choose from, each providing for a different level of protection. One important factor to keep in mind is the fact that the kind of license chosen will strongly depend on the specific business goals and on the valuation of the IPR by the company.

Still, adopting a Copyright Open Source IPR should not be considered more complex than adopting a Patent IPR.

No matter what, a "do-nothing" strategy can lead to disaster. In ***the Rambus v. Infineon*** case for example, JEDEC's IPR only covered issued patents. Then when the litigation occurred, the complainant argued that the intellectual property owner should be bound not only by JEDEC's express Intellectual Property policy on issued patents, but also by the unwritten understanding of all members that pending patent applications should be

[18] Article 9.1: The ownership of the copyright in STANDARDS and TECHNICAL SPECIFICATIONS documentation and reports created by ETSI or any of its COMMITTEES shall vest in ETSI but due acknowledgement shall be given to copyrights owned by third parties that are identifiable in ETSI copyrighted works.

[19] Article 11 Reproduction of Standards Documentation
MEMBERS may make copies of STANDARDS and TECHNICAL SPECIFICATIONS documentation produced by ETSI for their own use free of charge but may not distribute such copies to others.

disclosed. A clearer IPR policy would have helped to avoid that uncertainty and the related litigation.

Choosing a clear Copyright Open Source IPR is the solution adopted by the French CNRS[20], with INRIA[21] and CEA[22], have recently adopted the ***CECILL*** Free Software License[23].

3.2 Open Source IPR or Patents IPR

For free software proponents like James Clark, Richard Stallman or W3C supporters, patents IPR such as RAND or FRAND are discriminatory.

In particular, they feel it discriminates against open source software and the core principles of the open source philosophy. As Jim Bell writes about RAND in the W3C context: "RAND ignores the importance of open source. Open source is one of the most powerful, creative forces in computing today, and W3C Patent policy should facilitate it rather than blocking it via RAND royalty fees."

But it should be noted that not all Open Source advocates are against Patents IPR or RAND.

RAND licensing terms may be ***unavoidable*** or preferable to the alternative in situations where “IPR encumbrance” can become part of the technology scene. That is, accepting RAND license terms for a technology in a developing standard may be preferable to waiting until some new unencumbered technology is available.

As it is stated in one of the draft documents produced by a member of the IETF IPR Working Group called "Guidelines for Working Groups on Intellectual Property Issues": "It will always be better for the Internet to develop standards based on technology which can be used without concern about selective or costly licensing. However, increasingly, choosing an unencumbered technology over an encumbered alternative may produce a significantly weaker Internet. Sometimes there simply isn't any unencumbered technology in an area."

[20] CNRS: www.cnrs.fr
[21] INRIA: www.inria.fr
[22] CEA: www.cea.fr
[23] http:www.cecill.info

3.3 Open Source IPR and Patents IPR

Patents IPR may be acceptable by all and can be compatible with copyright Open Source IPR.

For example, the ***IBM statement on ebXML*** states that the Royalty Free Patent IPR license will be "free for all time": "Patents are granted and remain enforceable until they are either abandoned or until 20 years from the original application date". In such a case, the legal encumbrance is apparently reduced to a bureaucratic detail amounting to application for a license (with submission of a signed form) and waiting [an unspecified period of time] for a reply.

In addition, patents IPR like FRAND are an ***important instrument in negotiations between companies*** seeking to clarify their mutual obligations with respect to patented technologies. Even for OSS proponents, a FRAND license framework can be a good solution since, whilst being compatible with an Open Source License framework, it can provide a clearer picture of mutual obligations.

While some may strongly support Open Source only or Royalty Free only specifications in a majority of "open Internet" situations, it is unreasonable to believe that we know enough about the future to carve this in marble. The model explored by IBM is exemplary in this connection, explaining that standards consortia and corporations are better served by maintaining a policy of flexibility. Thus, the company considers royalty and patent issues on a case-by-case basis.

For example, FRAND licensing with royalty fees can be considered positive in some situations ***where the revenue generated can be used to financially support other software R&D*** governed by Royalty Free expectations. With such a policy it is common for large companies having significant software patent royalty revenue (IBM reported USD 2 billion revenue in 2002) to contribute substantial portions of its development to Open Source software and otherwise to license its experimental software for free (e.g., IBM alphaWorks and Eclipse).

To summarize, copyright Open Source IPR and patents IPR are not incompatible. Open Source Software follows simple rules leading to complex situations but it should not be seen as something un-understandable.
On the opposite, it should not be simplified as it seems to be the case with the Open Standards debate.

4 Open Standards

4.1 International Positions on Open Standards

New standardization models are emerging today, leading to new criteria and to a push for open standards: coming from the software side, some organizations like W3C are now promoting new forms of standardization, stressing on the need for Royalty-Free IPR and pushing for the recognition of open standards.

According to Ken Krechmer, there are 10 criteria to define an Open Standard[24] but in fact, there are now several definitions of Open Standards: BSA, France, Danish IT and Telecom Agency, IDA, ANSI, ITU-T, etc.

▶ *European Standardization Policy*

The directive 98/34 based on the article 95 of the Treaty aims at ensuring a smooth functioning of the Internal Market, by fostering transparency on national activities in the area of technical regulations and standards, and promoting the harmonization of such technical regulations and standards at European level. 98/34 EC contains a definition of a standard by stating that it must be "a technical specification approved by a recognized standardisation body for repeated or continuous application, with which compliance is not compulsory and which must be adopted and made available to the public." **As shown herein, no mention of the notion of "open standards" is made in the Directive, even if provision 24 mentions the need for openness[25].**

Directive 98/34 EC organizes an information exchange procedure on national standardization activities and provides the legal basis for addressing some requests called mandates to the European standardization organizations (CEN, CENELEC and ETSI). Member States have standstill obligations with regards to national standardization activities in those areas when European mandates have been entrusted to these organizations and the 98/34 committee provides follow-up on these mandates.

[24] The Meaning of Open Standards, Ken Krechmer, 2005.

[25] "Whereas the European standardisation system must be organised by and for the parties concerned, on a basis of coherence, transparency, openness, consensus, independence of special interests, efficiency and decision-making based on national representation."

▶ National debates on the definition of Open Standard

National governments enact Interoperability Frameworks listing recommended standards for e-gov activities[26].

Aside from the French position, UK, Germany, the Netherlands and Belgium drafted a series of frameworks for interoperability in which they give both a definition of Open Standards and a list of proposed standards complying with those definitions[27].

The German Standards and Architectures for e-Government Applications (SAGA) is described as the first step toward a common government set of IT-standards. Released in 2003 SAGA V2.0 defined a set of technical standards for e-Government integrated in a framework concept. The German Federal Government is now preparing SAGA V3.0, but these documents do not specifically address Royalty Free/FRAND or IPR issues[28]. They state the differences between standards but they do not make recommendations on licensing models. For example, while referring to the fact that ".png" is a picture format available under a very liberal open source license, ".gif" is still the preferred technology as long as ".png" is more commonly recognized by web browsers.

The Dutch position goes further, with a list of Open Standards[29] where it is specified that the costs for the use of the standard must be low and that when Intellectual Property would be present the standard must be made available on a royalty-free basis.

The French Parliament gave a legal definition to the notion of "standard ouvert" in June 2004: "any protocol for communication, interconnection or exchange, and any interoperable data format for which technical specifications are public, with no access restriction nor re-use restriction"[30].

Even though the definition is not crystal clear, some recent French legal developments do refer to "standard ouverts"[31] and there will certainly be

---

[26] But Venezuela decided to be more direct by legally imposing the use of Open Source software in order to promote a definition of Open Standards.
[27] eGIF in UK, SAGA in Germany, BELGIF in Belgium.
[28] Actually, SAGA V2.0 precises: "Some of the standards discussed are inseparably linked to licensed products. Our recommendation should be understood to be of a purely technical nature. Whether and at which conditions (single/group license) a product can be economically used must be checked from case to case".
[29] http://www.ososs.nl/
[30] LOI 2004-575 du 21 juin 2004, article 4 http://www.legifrance.gouv.fr/html/actualite/actualite_legislative/decrets_application/2004-575.htm
[31] The current definition is now beginning to be used in new laws and legal texts.

more in the future. All of them interpret this definition as being Royalty Free even if the text is not specific about it[32].

The Belgian Interoperability Framework (BELGIF)[33] approved a first list of standards on May 2nd, 2005. Clearly inspired from a similar understanding of what "open" means. The proposed process is quite original since the standards list is opened to contributions for 3 months. Anybody can propose changes and another list will be proposed during the second semester of 2005.

Then, on presenting "eNorge 2009 – the digital leap", Norwegian Minister of Modernization Morten Andreas Meyer declared in June 2005 that "Proprietary formats will no longer be acceptable in communication between citizens and government." As part of the plan, the Minister decided to write for the use of open source code in the public sector by the end of 2005 and to ensure that every part of the public sector in Norway use open source code and open standards by the end of 2006[34].

The same trends are growing outside Europe, where they are often mixed with other tendencies.

For example, the Chinese case illustrates that Open Source can also appear as a way to move around IPR in order to avoid the technological supremacy of dominant players. This is why China promotes Linux and other Open Source solutions on its government, trying to develop a local technology industry and their own standards[35]. Yet, the Chinese government has not taken any mandatory provision favouring open source solutions.

Following the same model, Brazil already tried to go one step further by imposing mandatory use of Open Source Software within administrations in 2003. The bill PL-2152/2003[36] concerning federal administration and public entities was voided, but a new proposal recently discussed to switch the project of federal government systems to Open Source Software. The project is currently under discussion.

---

[32] Today, this definition only apply to the information legally required to run a website. But since there is no contradictory point of view, "standard ouvert" is more and more considered as being royalty-free. It is also important to note that even restricted to information, this could lead to an important leverage effect since this information is mandatory for all.

[33] http://www.belgif.be/

[34] http://odin.dep.no/mod/norsk/tema/ITpolitikk/enorge/bn.html

[35] In august 2004 for example, the Chinese Ministry of Information Industry recently established MII the Open Source Software Promotion Alliance as a partnership of enterprises, non-profit organizations, representatives from NGO and individuals.

[36] http://www3.camara.gov.br/

And these trends are sometimes directly oriented towards standards. In 2003, the Japanese Ministry of Economy, Trade and Industry decided in to fund Japan OSS Promotion Forum with a dedicated working group on standardization and certification, in collaboration with a principle promoting cooperation and partnership amongst Japanese, Chinese and Korean OSS Promotion Consortiums[37].

Country	Date	Origin	Details
France	2004	Parliament	Legal definition of "standards ouverts"
Germany	2003	Federal Government	Interoperability Framework
Belgium	2005	Government	Wiki Interoperability Framework
Denmark	2004	ICTU	Interoperability Framework
China	2004	Ministry of Information Industry	Established MII the Open Source Software Promotion Alliance
Norway	2005	Minister of Modernization	Call for a plan to use Open Source Code and Open Standards by the end of 2006
Brazil	2003	Parliament	Imposing the use of Open Source within administrations
Japan	2003	The Ministry of Economy, Trade and Industry	Investing 1 billion yen to help the private sector to adopt open source platform with the creation of the Japan OSS Promotion Forum with on standardization and certification working group
ITEMS International			

37 http://www.ipa.go.jp/software/open/forum/Contents/StearingC/sapporo_kuwahara_presen_enV5.pdf

▶ EICTA Definition of Open Standard

EICTA has its own definition of Open Standards[38] along the following lines:

- the evolution of the specification should be set in a transparent process open to all interested contributors (criteria of control)
- the technical requirements of the solution should be specified completely enough to guarantee full interoperability (criteria of completeness)
- there is a substantial standard-compliant offering promoted by proponents of the standard (criteria of compliance)
- fair reasonable and non-discriminatory access is provided to intellectual property unavoidably used in implementation of the standard (criteria of cost)

▶ ITU Draft definition of Open Standard

In March 2005, the ITU TSB Director's Ad Hoc group also discussed on the definition an "Open Standard" from the perspective of standardization work in ITU, but they only wrote a preliminary definition. As mentioned, even though it does not constitute an ITU-T definition, it may be useful to help progress the discussions elsewhere than in ITU[39].

Following their point of view, ITU-T has already a long history of open standards development, defined as[40]:
(*"Open Standards" are*) standards made available to the general public and are developed (or approved) and maintained via a collaborative and consensus driven process. "Open Standards" facilitate interoperability and data exchange among different products or services and are intended for widespread adoption.

Other elements of "Open Standards" include, but are not limited to:

- Collaborative process – voluntary and market driven development (or approval) following a transparent consensus driven process that is reasonably open to all interested parties.
- Reasonably balanced – ensures that the process is not dominated by any one interest group.
- Due process - includes consideration of and response to comments by interested parties.

---

[38] See EICTA Comments to IDA's European Interoperability Framework http://portal.etsi.org/docbox/workshop/SOS_Interoperability/Background_papers/EICTA%20comments%20to%20IDA%20European%20Interoperability.pdf

[39] This definition was rencently considered finalized by the ITU-T. It was approved by the IPR Ad Hoc Group and will be presented to TSAG for information

[40] http://www.itu.int/ITU-T/othergroups/ipr-adhoc/openstandards.html

- Intellectual property rights (IPRs) – IPRs essential to implement the standard to be licensed to all applicants on a worldwide, non-discriminatory basis, either (1) for free and under other reasonable terms and conditions or (2) on reasonable terms and conditions (which may include monetary compensation). Negotiations are left to the parties concerned and are performed outside the SDO.
- Quality and level of detail – sufficient to permit the development of a variety of competing implementations of interoperable products or services. Standardized interfaces are not hidden, or controlled other than by the SDO promulgating the standard.
- Publicly available – easily available for implementation and use, at a reasonable price. Publication of the text of a standard by others is permitted only with the prior approval of the SDO.
- On-going support – maintained and supported over a long period of time.

▶ *IDABC Definition of Open Standard*

Another definition of "Open Standards" was given in the November 2004 by IDABC in the European Interoperability Framework (EIF) for Pan-European e-Government Services [41], Final Version 1.0. According to EIF, open standards are:

- Adopted and maintained by a not-for-profit organisation
- Ongoing development must occur on the basis of an open decision-making procedure available to all interested parties (consensus or majority decision etc.)
- Published
- With the standard specification document available either freely or at a nominal charge.
- It must be permissible to all to copy, distribute and use it for no fee or at a nominal fee.
- The intellectual property – i.e. patents possibly present – of (parts of) the standard is made irrevocably available on a royalty-free basis.
- There are no constraints on the re-use of the standard.

[41] See EC announces Open Standards Definition at Dutch Presidency Conference http://europa.eu.int/idabc/en/document/3492/469

It must be noted there that the EIF document's "minimal characteristics" necessary for consideration as an "open standard" are largely identical to or comparable with other definitions of "open standard" proposed by members of the Open Source community such as Bruce Perens.[42]

Specifically, the provision that the intellectual property inherent within and bound to a published standard should be made "irrevocably available on a royalty-free basis" is argued to be a natural and unavoidable characteristic of the Open Source model.

It must also be noted that this definition for Open Standards is limited at two levels. On the one hand, the IDABC document only intends to address the e-Government sector. Second, it only addresses some fundamental norms within this specific sector. This limitation is key since IDABC claims that these recommendations only apply to a small number of norms within the e-Government sector.

Still, these various attempts to define “open standard” are triggering debates on a series of issues that no SDO can leave aside. One can reasonably surmise that there will be other attempts and subsequent debates on those topics in a very near future.

[42] It mirrors language in the much-cited "Open Standards: Principles and Practice," by Bruce Perens as well as the draft "Principles for Open Source and Open Standards" formulated for the Open Standards Alliance.

▶ *Open Standard and Policy*

As discussed previously, Directive 98/34 EC does not specify what an open standards is (yet provisions for openness), and IDABC's definition does not intend *per se* to go beyond the e-Government sector or to be a reference for the overall EU ICT standardization policy.

With this important reserve, it should be noted that Directive 98/34 EC is under revision and it is likely those debates will influence policy makers, even if current discussions at EU level do not seem to include debates on openness and/or IPR policies for ICT standards.

Other stakeholders such as EICTA support the viewpoint that standards ***should*** be voluntary and that standardization efforts should be driven by industry.

We believe that so far, the proposed definitions of Open Standards wherever they come from are trying to solve a problem of standard policy by enacting a new IPR policy when in fact Open Source and Open Standard are two different things.

Open Source is a form of IPR policy that has been successful in some standardization sectors but it is not meant to be a standardization policy, and the two things should be clearly distinguished if one wants to learn from each other.

4.2 Open Source and Open Standards are different problems

As previously discussed:

- ***There is not a single Open Source model,***
- Open Source licenses follow simple rules leading to complex situations,
- Patents IPR and Open Source IPR are not incompatible under a pragmatic approach.

As Douglas Heintzmann from IBM puts it: “While the roles of standards and open source software do overlap in that many companies who are using open source in part as a means of implementing open standards, ***the cores of their respective value propositions are distinct*** and should be discussed as such.”[43] For IBM, “The similarity to open standards lies in the development-by-community approach, which makes the process visible to all interested parties. Open standards complement software development generally, and may be applied to projects that are ultimately released as open or closed systems.”[44] **The goal is to cooperate on standards and compete on implementation**[45].

The IDABC’s Interoperability Framework was designed to optimize interoperability especially in terms of allowing and encouraging the use of open source software. It is not surprising, therefore, that the "minimal characteristics" necessary for consideration as an "open standard" were written to be compatible with the dominant business models, licensing practices, and code redistribution requirements of the Open Source community. That would typically exclude royalty payments for Patented IPR that involves patent licensing under a required executable license.

Current debates are fuelled by mixing the concepts of open standards and open source and they slide between the two notions. Open Source, patents IPR and Open Standards have common points but they address different issues and without a proper analysis, their assimilation can lead to difficult consequences for standardization.

---

[43] An introduction to open computing, open standards, and open source, Douglas Heintzmann.
[44] The open standards imperative: IBM’s open-for-business strategy.
[45] The Business Software Alliance emphasized the distinction further. In its February 2005 Statement on Technology Standards, the BSA stated: “While an open standard is a technical specification, open source software (OSS) is software that may be used to implement an open standard in a particular product or service. […] Some open source projects are closely associated with particular open standards (e.g., Apache with HTTP, or MySQL with SQL), and some standards even choose to release their reference implementations under open source licenses. However, the mere availability of source code is neither necessary nor sufficient to make something a standard, much less an open standard.”

4.3 THE RISKS OF MISUNDERSTANDING

Below are three examples showing the ambiguities stemming from dealing with the notions of Open Standards and Open Source side by side and the need for clarification.

▶ SOAP

When IBM decided to support ***the development of Simple Object Access Protocol*** (SOAP) 1.1 at W3C, it initially provided the source code to co-develop the specification.

As work progressed, IBM committed to build tools to support developers who wanted to adopt the standard. Within two days of the release of the SOAP 1.1 specification, IBM released the SOAP for Java (SOAP4J) toolkit to the community at large. Within two months, SOAP4J was ***released as open source code*** to the Apache community. Finally, SOAP support was ***incorporated into the IBM WebSphere*** Application Server (a derivative work from the Apache Server) as a first-to-market commercial offering.

▶ JAVA

The example of Java is a different example of possible misunderstanding between Open Source and Open Standards.

Sun Microsystems created Java. Since February 2000, all Java Enterprise Edition related specifications were created and developed through the Java Community Process (JCP) where most of the Java Specification Requests (JSR) were submitted by other companies.

But the company wanted to avoid incompatible implementations. Thus, it retained control over Intellectual Property of these specifications under the related licence to the java community process: the SCSL[46].

In December 1999, Sun Microsystems announced the release of the Java 2 Enterprise Edition (J2EE). The code of the J2EE extension code was developed through the JCP. Sun was the original contributor for the majority of this code.

[46] To be precise, the specifications were under a standard specification license which said that they were available for implementation provided that it fully implements them, does not sub- or superset the API's namespace and passes the TCK. The copyright of specifications was owned by the Spec Lead of a JSR. Sometimes Sun sometimes other (including IBM). The SCSL is the license under which Sun's own Java EE implementation source code was made available at the time. Since June 2005 this source code is now licensed under CDDL, an OSI-approved open source software license.

Then, Sun Microsystems begun negotiating with companies using the J2EE standard in their software: it wanted to ensure compatibility, and ask for the payment of royalties in order to offset the cost of building, maintaining and supporting this suite or “standard of compatibility”.

From their point of view, the value proposition of Java is the portable and interoperable aspects of the technology. At the same time, Sun Microsystems purchased iPlanet, a competitor with IBM's WebSphere.

But IBM who contributed to 80 percent of the code refused to pay the royalties. At that time they complained that Sun Microsystems was using its control of a standard to hinder competition. They also advocated that they would toughen the compatibility tests in order to block competition while acquiring market shares for iPlanet[47].

▶ *Sender-ID*

Finally, the issues on reconciling Open Source and Patent Licenses embedded in standards are exemplified in the SenderID case where ***the IETF's MARID Working Group dissolved*** by the Internet Engineering Steering Group. Workgroup members were unable to reach technical on the specification.

In its IPR statement to IETF Microsoft advocated that the "Microsoft Royalty-Free Sender ID for E-Mail Specification License Agreement" met the formal requirements of the IETF for declared Intellectual Property. It permitted members to license its essential patent claims on a RAND basis without royalties, but a significant number of Open Source members disagreed, including the Apache Software Foundation and Debian[48] who argued that they could not implement or deploy the IETF Sender ID specification under ***Microsoft's RAND license terms***. Although royalty payments were not at issue, they felt that several other standard RAND license terms were declared to prevent implementation of the Sender ID specifications in open source contexts.

Apache stated in an open letter that Microsoft ***Royalty-Free Sender ID Patent License Agreement terms (were) a barrier*** ”to any ASF project which intends to implement Sender ID. We believe the current license is generally incompatible with open source, contrary to the practice of open

[47] IBM has asked to add the following comment: "IBM believes Section 4.3 on JAVA is significantly misleading and should not be considered in the use of this document."
[48] Debian is a widely used distribution of free software that is well known for its high quality and its adherence to the Unix and free software philosophies.
http://www.debian.org

Internet standards, and specifically incompatible with the Apache License 2.0. Therefore, we will not implement or deploy Sender ID under the current license terms." Apache identified several incompatibilities between the zero-royalty RAND license and open source software licensing[49], but ***its main key issue was that Microsoft's License included "non-sublicenseable" Patent license terms***[50].

Based on feedback[51], Microsoft proffered an amended version of its Royalty-Free Sender ID Patent License Agreement to address such concerns[52].

Because the MARID Workgroup members were not reaching technical consensus on the Sender ID specification, the Internet Engineering Steering Group area director dissolved the group.

[49] As many Open Source licenses allow users to sublicense derivative works, they would have been incompatible with Microsoft's RAND license. When working with Open Source or Free Software projects, the question of compatibility of licenses is often of great importance.

[50] http://xml.coverpages.org/Apache-SenderID.html

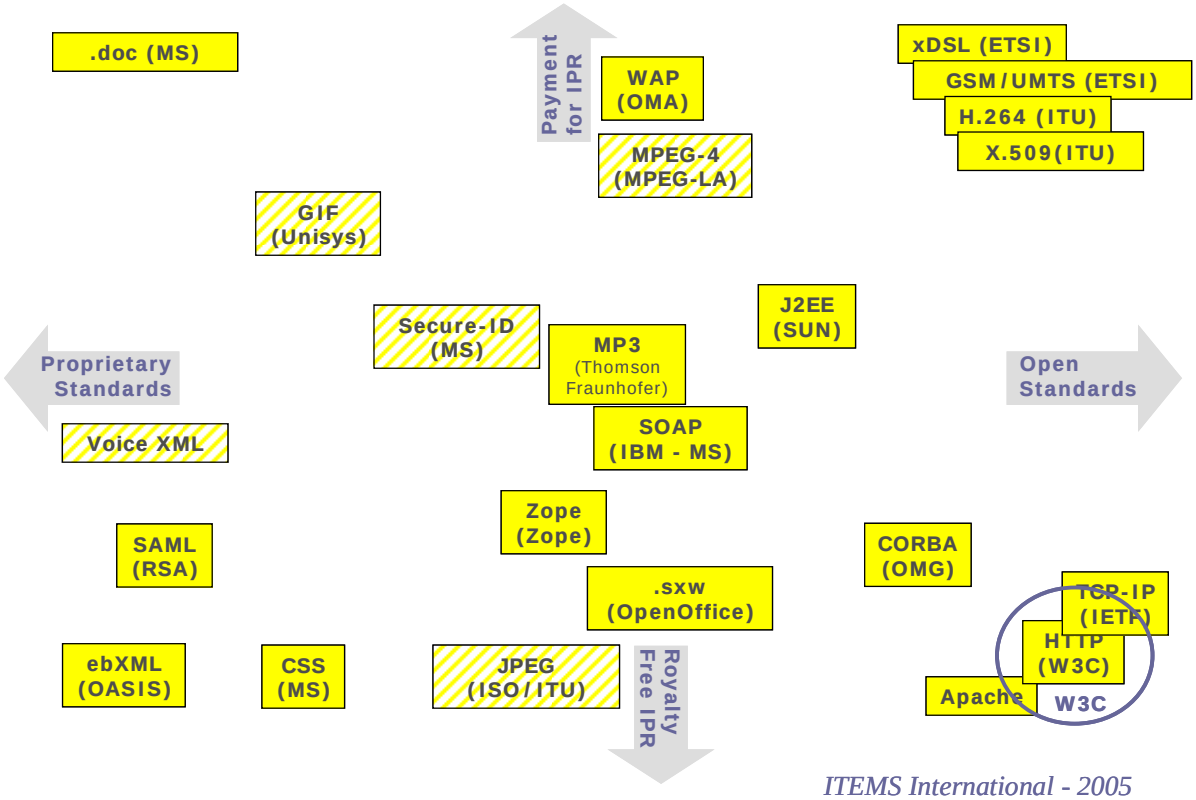
[51] For example, Larry Rosen, general counsel of the Open Source Initiative, mentioned in his 2005 book, Open Source Licensing: Software Freedom and Intellectual Property Law, that a "non-sublicenseable" patent license term does not necessarily make the license incompatible with open source. He also called the W3C Royalty-Free License a "model for open standards patent licenses compatible with open source," even though it includes terms unfriendly to open source such as "non-assignable" and "non-sublicenseable" provisions.

[52] http://www.imc.org/ietf-mxcomp/mail-archive/msg03500.html.

4.4 TENTATIVE MAPPING: CROSSING OPEN AND OPEN

The complexity of previous examples shows that standards cannot be sorted out in a simple table like Open Licenses or SDO's IPR policies. On the one hand, it is important to proceed standard by standard rather than a classification by standardization bodies or companies. On the other hand, standards (de jure or *de facto*) should be sorted out between Open and Proprietary standards as well as between Licensed and Royalty-Free.

The mapping below proposes to show the diversity of situations in combining theses two approaches[53].



ITEMS International - 2005

[53] The idea here is not to be exhaustive or to launch a debate on the precise position of each standard. It's just to represent the diversity and complexity in a heterogeneous panel of standards.

5 The Impact of Open Source on Standardization

5.1 When software changes the rules

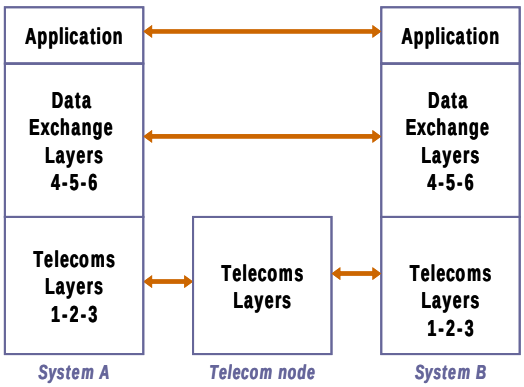
A series of factors including convergence and the increasing role of software are affecting standardization models. ICT standardization is moving from a limited series of players specifying end to end systems (from infrastructures and up) towards a services-based heterogeneous environment, typically software and computer-oriented.

Those alternative models of standardisation complement and sometimes challenge or supersede the formal standardisation model: private sector standards consortia, global partnerships of private sector players now regularly deliver “standards”, not to forget de facto proprietary “standards”.

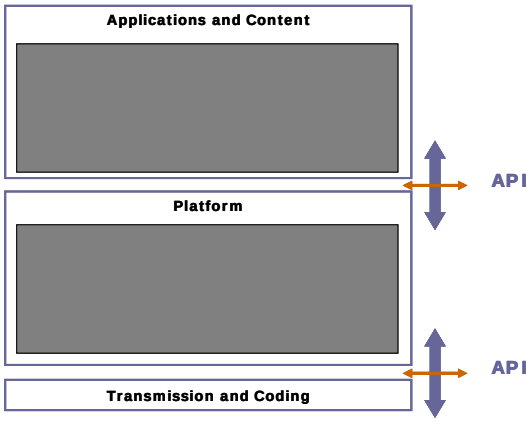
▶ *Cultural variants in approaching convergence*

Whenever convergence is at stake, three very different cultural patterns are in the ICT standardization arena.

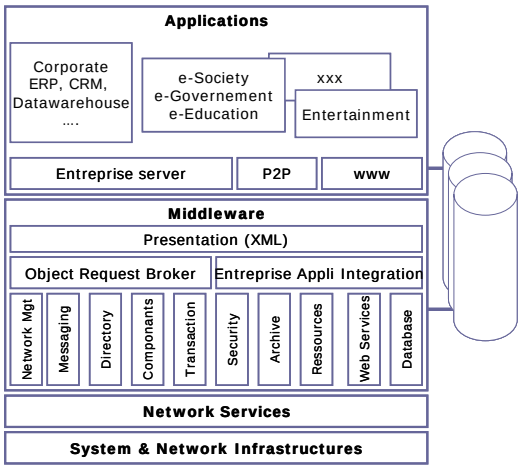
- The Telecom culture, based on formal layers including detailed protocol and services. In this model, everything is oriented towards communication and historically there is a powerful culture of formal description of services and protocol standards. The application layer is one among others and doesn’t differentiate system application from end-user applications.



- The Broadcast culture, based on black boxes strictly controlled by vendors. The issues of standardization are critical about formats and coding. But standards are not so developed in the intermediate layers where only a few APIs (Application Program Interfaces) are under discussion.

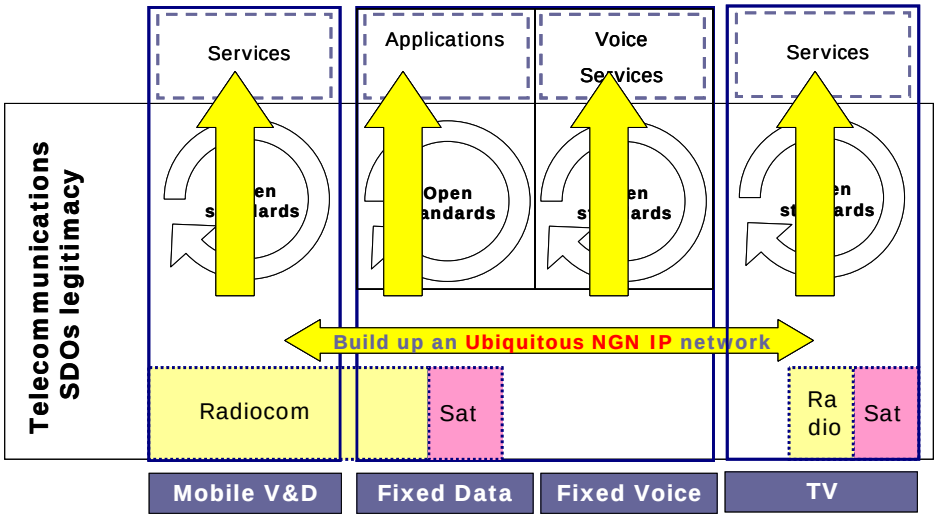


- The IT culture, based on dynamic components and interfaces able to adapt quickly to market trends. The concept of standardization is less clear or formal than it is in telecommunication. Vendors try to promote a software component or a product as a standard. The distinction between interfaces and software product is often ambiguous.



ITEMS International - 2004

Today, standards-makers coming from those different cultures must collaborate, even though their approach to priorities for convergence may differ.
In Telecommunications, Convergence means in priority “To build up an ubiquitous NGN network for multimedia fixed and mobiles services”.



ITEMS International - 2004

In TV, the priority is to develop universal and secured platform in order to control many kind of services and content. In IT, middleware is the key of convergence for a global architecture able to support different kinds and interoperable applications and contents.

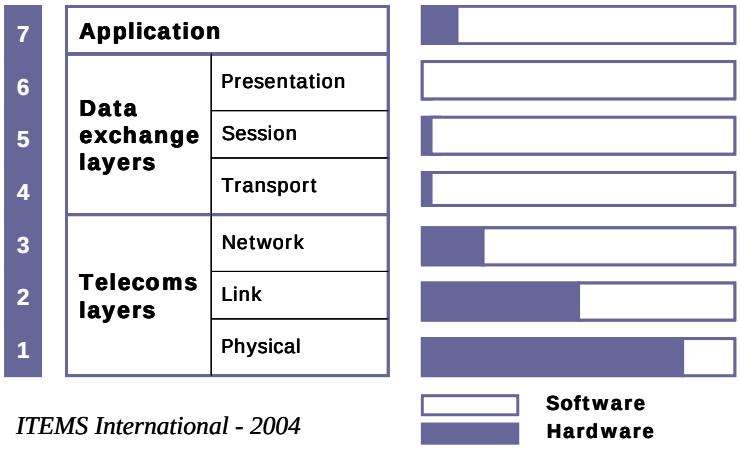
Priorities for convergence

Telecom	IT	TV
Ubiquitous NGN IP Network for Multimedia Fix / Mobile services	Control and Management of applications and services on a Global Middleware Platform	Universal Platform of content and services delivery

ITEMS International - 2005

▶ Software leads the game

The fact of the matter is that across the ICT sector, software is now prevalent in the development of products. In the case of Telecommunication sector, one can illustrate the share of software in the different layers:



ITEMS International - 2004

This means experts working on the developments of communication systems are essentially software developers. Tools, methods, language and culture are related to software background.

The rules driving the software sector are becoming prevalent as well. Software components are spread on the market very quickly. The priority is not to standardize first, but to be the first on the market. With software pervasiveness standardization is increasingly defined ex-post rather than ex-ante.

Naturally, everything is not so simple: vendors have to analyse their chance 4to go alone or to gather forces. But the trend is clearly that software gives players the opportunity to propose in a dynamic process:

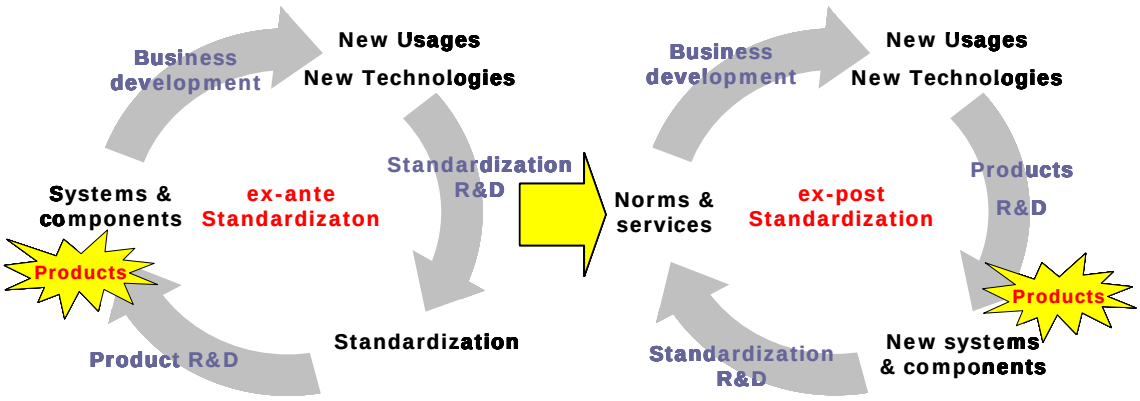
- specifications for new interfaces,
- softwares that describe and implement these interfaces.

The beauty of it all is that today, a standard can be defined as “Open” when it is actually a proprietary piece of software.

▶ From ex-Ante to ex-Post Standardization: the key role of software

This view is becoming well accepted as the role of software increases and as the key factor for convergence resides in what is known as middleware.

Since software standardization works with a logic of components, standards are mainly defined ex-post from components available on the market.



ITEMS International - 2004

Ex-ante standardization has proven effective when hardware was prevalent in ICT. It was possible however very risky for industrials to develop products without a minimum set of standards. In a software-driven ICT world, it would be nonsense to say the risk does not exist, but the risk-cursor has moved.

This explains why a plethora of organizations and fora emerge: in order to deal with life on life's terms (speed and stability). Fora allows vendors to gather forces in order to be as fast as possible and put products on the market before competitors. With fora, players manage a compromise in gathering:
1- representative players to be legitimate on the market,
2- a compact kernel of these players in order to get results quickly.

In this way, since standards and profiles are important in fora, the priority is to be able to propose products as fast as possible. The next step is to find a SDO in order to formalise and ratify the standard de facto defined (from de facto to de jure).

Depending on the strength of the fora, it can be generally an interesting opportunity for a SDO to deal with. That reinforces its own legitimacy on the standards market place.

It should be noted as a side comment that ex-post standardization in a software-driven ICT industry leads to multiple questions with regards to public policy and Industrial Policy.

5.2 OPEN STANDARD: A DEBATED NOTION

▶ "Open standards" under discussion

Originally, the term "open" meant open participation by all interested parties and deliverables accessible to all interested parties during or after standards elaboration. Proponents of new definitions may not agree with those long involved in the standards industry, even with those that strongly support open source development and licensing.

Equating patents and standards is just a misunderstanding since patents and standards serve different (sometimes conflicting) needs. If not all, many standards organizations have a patent policy because they must clarify their IPR rules[54].

In this particular case, SDOs can choose either option:

- Developers of the standard start over again so that standard with its encumbered patent exists for a period of time as short as possible[55].
- The patent holder gets mauled by the developers of the standard[56].

In this situation, patents are like a ticket to negotiate licenses in the standardisation process, but the same can be said of any other IPR like copyrights: patents may cause potential problems, they are simply another form of IPR than copyright.

It is also interesting to ask why the debate on open standards is taking place today. The issue seems closely related to the competition on the market between standards bodies and fora, and the pressure from public authorities to use fora outputs in addition to "formal" standards bodies' for fulfilling public policy needs. In this debate, a number of informal standards setting

[54] For example when dealing with the situation of discovering a patent in the middle of the developing specification process

[55] While Microsoft actually went out of its way to try to do the right thing in the SenderID debate last summer, the standards community's response was direct and undeniable when they even thought they were at risk.

[56] An exemplary situation is when Northern Telecom attempted a strong arm tactic with patents against a particular VITA effort: the VITA members organized legally outside of the standards organization and won.

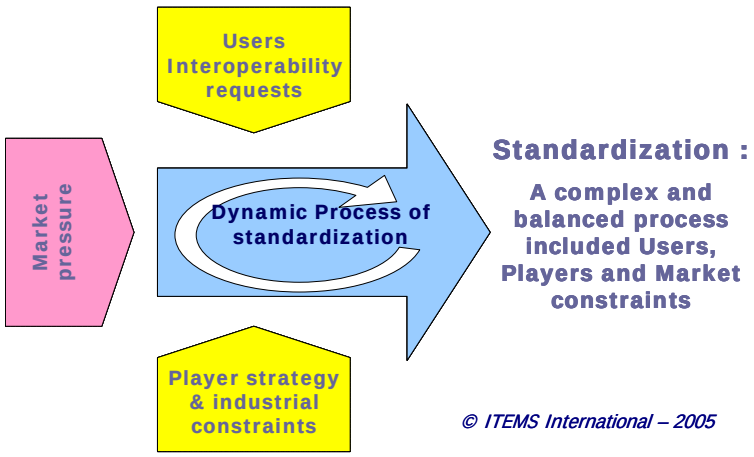
bodies and trade associations propose to erase the distinction between official standards and industry fora specifications[57].

▶ *The increasing role of users*

These debates are also triggered by the increasing role of users[58]. When needed for a specific usage, standards can be chosen in many different organizations. The key issue of openness and interoperability comes in a different flavor.

Such is the case of governmental organizations (national administrations or local authorities) which are generally huge and complex. They do not represent necessarily an homogeneous organization, but a grid of multiples entities. In the case of a major city, the local authority needs to manage relations with national or regional administrations, local schools, libraries, social and health care local administrations, emergency organizations, etc. In the case of schools, they can be in charge of purchasing the IT equipments. There is an effect of scale in administrations that emphasize the interoperability issue.

With the development of open source and the availability of Royalty Free standards, those organizations are not precise enough on what open standards are, the difference between open standards and open source and why they should be interested. In such a context, it comes as no surprise that the general understanding of "open Standard" is to open the code, or in other words: open source[59].



[57] Stating that the technical requirements of the solution should be specified completely enough to guarantee full interoperability.
[58] "Users" refers here in priorities to organizations (or their representatives) who buy technologies and systems for an usage (and not to an organisation who uses specifications to setup a system).
[59] Brazil is a very striking example of this choice now embedded in public ICT policies.

What remains is that users express mainly their concerns on uppers layers (applications and services). But with the increasing role of open source, they have had the opportunity to put pressure as to redefine the rules across the layers. IDABC e-Link is a good example of those interoperability issues across the layers.

▶ *Applying modern openness to standardization*

The IDABC definition of open standard embedded in the European Interoperability Framework raises legitimate concerns for a number of standards organizations.

As a matter of fact, this definition of open standard is probably what can be expected to be formulated with increasing frequency by advocates of the open source software community. Moreover, the problem is not likely to get solved by itself as there seems to be a growing number of open source involvements in the standardization process: Matroska / MP4, Zeroconf / RendezVous, etc.

In addition, “Open Source” is trendy, especially when mixed with “free” because it offers a way to play a role for non commercial entities.

As of today Open Source and Open Standards both suffer various definitions and relate to different problems that cannot easily be linked together without a proper typology and analysis.

This does not mean that IPR rules are irrelevant for Open Source Software. Standard Setting Organizations should rather improve their use of copyright rules. In doing so, they can avoid potential hold-ups by Members of the organization, while at the same time ensuring that Intellectual Property owners will be appropriately rewarded with reasonable royalties when needed.

It seems important to establish a crossed typology of IPR Policies, Open Standards and Open Source Licenses in order to know how to make them work together on a pragmatic basis rather than relying on Free Software advocacy.

5.3 THE NEED FOR A CLEAR IPR POLICY

Standards involving network interoperability for the World Wide Web raised a variety of new challenges. Today, new IPR should consider the development dynamics of Web standards and the environment in which they have been implemented over the time.

The key challenge posed by Intellectual Property in any standards arena is that participants in a standards body will be unwilling and/or unable to work collaboratively if, at the end of the process, the jointly-developed standard can only be implemented by meeting licensing terms that are unduly cumbersome, unknown at the beginning or the end of the process, or considered unreasonable. At the same time, many commercial and academic participants in standard processes are unwilling to waive all of their Intellectual Property claims at the door of the standard group.

Therefore careful policies are required to establish whether, and on what terms, patent claims may be used by those who participate in the design of standards.

▶ *W3C: The difficulty to evolve when Open Source and Open Standards get together*

Open source implementations have played an essential role in the development of many of W3C standards that were implemented early with open source code, enabling large numbers of developers (commercial and non-commercial) to incorporate new Web features into software. They were then writing without having to develop those features from scratch. Finally, the diversity of contents represented by the over two billion Web pages is only possible because their creators were able to use key standards such as HTML and CSS.

In the same way, XML is a set of standards for structured data on the web that was basically set out in three months of feverish work by 11 very different individuals.[60]

Responding to increasing concerns about patents, W3C created the Patent Policy Working Group (PPWG) in July 1999 to forge a patent policy.[61] The

[60] Arbortext - Eve Maler, Dyansoft - Steve DeRose, Hewlett-Packard - Dave Hollander, Independent - Elliot Kimber, Independent - James Clark, Microsoft - Jean Paoli, NCSA - Tom Magliery, Softquad - Peter Sharp, Sun - Jon Bosak, Textuality - Tim Bray, Univ. Illinois - Michael Sperberg-McQueen

[61] Participants in the Patent Policy Working Group included representatives from AOL Time Warner; Apple; AT&T; Avaya; Daisy Consortium; Hewlett-Packard Company; IBM; ILOG.; Intel; Lexmark; Microsoft Corporation; MITRE; Motorola; Nokia; Nortel Networks; The Open Group; Oracle Corporation; Reuters, Ltd.; Sun Microsystems; Xerox Corporation; as well as

Patent Policy Working Group's first proposal suggested a two-track approach to patent policy at W3C, allowing both RAND and Royalty Free licensing modes. According to the proposal, each time a new Working Group was formed to develop a standard, a choice would be made whether the standard would be developed to be implemented on a royalty-free or a RAND basis.

Responses to that draft were strong, both from W3C Members and the public. Support from W3C Members was mixed, and reviewers sent detailed comments. Public reaction to the draft was negative, primarily because the framework would allow W3C to include in its Recommendations technology known to be patent-encumbered, and that implementers might therefore have to pay a license fee to implement a W3C Recommendation. The strongest reaction came from communities of open source developers who declared in several thousand e-mails sent to the W3C public comment mailing list that a RAND approach would cause open source developers to stop using W3C web standards[62], some even estimated that it would constitute a take-over of the Web by large corporate interests.

W3C responded to this negative reactions by adding invited experts to the Patent Policy Working Group to represent the Open Source community, creating a task force within the Patent Policy Working Group to examine how to accommodate the Open Source community, and creating a public home page for the PPWG, publishing public meeting records, and committing to regular public circulation of new drafts before the policy was finalized.

There is much to learn about how the Royalty Free licensing model function on the web standards environment. What is clear is that the vast majority of this technical community believes that it is essential to make this policy work.

Actually, this debate over the policy clearly shows that continued innovation and advances depend on the quality of the chosen IPR policy.

A similar debate is now beginning around OASIS.

---

invited experts from the Free Software Foundation, Software in the Public Interest, and the Open Source Initiative.

[62] Which would forbid some Open Source implementations. For example, without specific provisions, it would be impossible to create GPL software implementing such RAND specifications since RAND would forbid the application many copyleft articles from the GPL.

▶ *OASIS: The pressure to assimilate Open Standards to Open Source principles*

The newly released OASIS IPR Policy demonstrates the new situation for standards organizations[63]. Offering a more detailed choice than the W3C proposed policy, it allows negotiation on a case-to-case basis in order to force the individual technical committees to discuss about IPR before anything else, and thereby obligates them to the course chosen by all.

Members must declare patent applications as well as awarded patents, so everything is known well in advance within the group. When they withdraw, they must keep ensuring their licensing requirements, preventing patent abuse. Then, whether they opt for RAND or Royalty Free, open or proprietary IPR, all of this is the result of a choice from players.

That did not avoid open source proponents to call for more openness, writing a protest letter regarding the debate surrounding document format standards prepared within OASIS[64]. The debate now looks much like the one that occurred around W3C, mixing interoperability, open or proprietary IPR comparisons with Royalty Free or RAND debates.

[63] The new OASIS IPR Policy is effective as of 15 April 2005. As explained on the OASIS website, Technical Committees formed prior to that date will be required to move to this Policy by 15 April 2007, in the manner provided in the IPR Transition Policy. However, committees that have not yet transitioned will continue to operate under the terms of the legacy OASIS IPR Policy. Of course, the version of the Policy applicable to each committee and its work product will be clearly marked on the public home page for that committee. http://www.oasis-open.org/who/intellectualproperty.php

[64] See A Call to Action in OASIS http://perens.com/Articles/OASIS.html

▶ *Microsoft: Other solutions*

For Microsoft, an open standard is only a technical specification (i.e., a set of technical instructions and requirements), a practice or a reference model. It is used to define how two or more products or services can interact in some predictable fashion and to ensure that certain minimum requirements are met. Common examples include SQL, HTML, TCP/IP and programming languages like C/C++. Standards allow multiple competing implementations that can be judged in the marketplace to make sure that certain basic requirements are met. Vendors may then choose to build beyond the standard in order to establish competitive differentiation on the market.

By contrast, Microsoft generally defines an open standard as a standard that has the following characteristics:

- Developed, maintained, approved, or affirmed by consensus, in a voluntary, market-driven standards-setting organization that is open to all interested and qualified participants
- Published without restriction (in electronic or tangible form) in sufficient detail to enable a complete understanding of the standard's scope and purpose (e.g., potential implementers are not restricted from accessing the standard)
- Publicly available without cost or for a reasonable fee for adoption and implementation by any interested party; and
- Any patent rights necessary to implement open standards are made available by those developing the standard to all implementers on reasonable and non-discriminatory (RAND) terms (either with or without payment of a reasonable royalty or fee).

In Microsoft's view, customers choose the product that does more than simply meet their base requirements. It is this relationship between specification implementation, and competitive differentiation that provides basic interoperability between vendors, drives competition and spurs innovation.

The examples above show that definitions of open standards can vary widely between different organisations: from one model to a completely different one, from the Open Movement definitions to leading standards development organizations. Following this, it should be important to take some time in order to raise some trends for the future.

5.4 WHAT'S NEXT: TRENDS FOR THE FUTURE

Forecasting is a risky game, but we will try to describe some scenarios of how the support of the EU policy makers and national governments can impact the future of open source within standardisation. The versatility of the approaches, and the great mobility of the ICT environment makes it difficult to do "crystal ball' predictions, and in addition, the open source movement has already proved how it can progress by itself.

First of all, we tend to think that the importance of software has not yet reach its full scale. It will still extend much further, imposing its models to ICT.

This leads to define Open Standards with regards to this development, taking into account the need for interoperability and the specificity of software models. This would especially have an impact on the importance of open process and Intellectual Property policies. It will be important to point out that financial issues are not as important as Intellectual Property issues.

Standardisation could also embrace the Open Model and study the solutions it has to propose. For example, double licensing would allow ensuring specific IPR and financial conditions for e-gov services, or FLOSS projects.

▶ *Improving Licensing in IPR Policy*

As Open Source is considered by some players and users to promote "Open Standards", it is important to point out the differences between Open Standards and Open Source.

Open Source usually refers to a method for licensing software based on copyright in which the source code is available to the general public for use, modification and distribution. Open Source is only an IPR policy. Open Source Software is not an open standard *per se* and it does not automatically provide interoperability.

Allowing Open Standards to be assimilated with Open Source principles would lead to the development of Open Source Software implementations rather than the development of Standards. However, Open Source Software is a topic of major importance for e-gov and software players, both as a strong development model and as an example of good IPR policy with regards to the public. In this case, several points need to be addressed.

- First, Open Source cannot guarantee interoperability. It is crucial to point out what are the additional IPR rules that would foster interoperability and to define a clear IPR framework. For example, the

IPR rules for Open Standard should be different when based on copyright, on patents or on a mix of both. In any case, IPR rules should be clearly defined to avoid

- For players, legal and business uncertainty.
- For players and Governments, without clear IPR regulations, could use Open Standards to impose their own agenda.
- For the public, there would be no guarantee of interoperability.

- Second, the development model chosen by IETF or OASIS would lead to think that a consensus on clear and transparent IPR rules is more necessary to any standardization process than a given development model, open or proprietary. The Commission already published such document in 1992 with its Communication entitled "Intellectual Property Rights and Standardisation"[65]. It considered the use of industry standards and concluded that there was a public interest in promoting both intellectual property rights and the use of industry standards. It recommended that democratic and pro-competitive processes were followed in drafting standards and setting out a set of "best practices" guidelines, as follows:
  - Standards should not include intellectual property right whose owner is not prepared to license.
  - Standards must be available for use on fair, reasonable and non-discriminatory terms from intellectual property right owners.
  - The standards body and intellectual property right holder should each use their best efforts to identify relevant intellectual property rights applicable to any standards.
- Nevertheless, this document can only be seen as a first step pointing out the importance of the topic and several other points still need to be precised. First, it would be important to make a difference between Standards following the clarity of their IPR policy. Moreover, Standards should not only be compared with regards to their technical specifications or following their development model.They should also be compared on the content of their IPR policy.

When defining open standards, a balance between standardisation and intellectual property rights has to be struck. Key to achieving balance seems to be:

- **IPR Transparency**, the existence and identification of intellectual

[65] Commission communication on Intellectual Property Rights and Standardisation, COM(92) 445 final, 27 October 1992.

property rights should be known before the standard is voted upon.

- **IPR Non-discrimination**, the insurance that licenses are going to be available and the detailed conditions.

▶ *Avoiding the frontal debate FRAND vs. Royalty-Free*

It is certain that Open Standard must take care of the rights of intellectual property rights holders.[66] As explained above, any definition of open standard adopted will have consequences for the organisation adopting the definition but it will also impact those who interact with it.

In that sense, excluding non-Royalty Free technology could mean to force them to give up their rights. Such would not only be contrary to the overall policy of the Commission to strengthen intellectual property rights protection, but could moreover be contrary to the EU obligations under the WTO Agreement on Trade-related Intellectual Property Rights[67].

On the other hand, choosing a strict RAND policy would forbid some Open Source implementations[68]. Any chosen definition will have a broader impact than only pushing for RAND or Royalty Free.

[66] And it must be noted that Open Source principles are based on IPR legal provisions.
[67] Especially article 31 of the TRIPS agreement which provide specific provisions in case of compulsory licenses.
http://www.wto.org/english/tratop_e/trips_e/t_agm3_e.htm
[68] Strict RAND policy would especially forbid the development of GPL implementations.

▶ *Qualify the context and field when defining Open Standard definition(s) is relevant*

The IDA proposed definition of open standards for the European Interoperability Framework is not intended to be applied outside its specified area of validity.

Also, there are principles limiting the discriminatory effect of the OSS-styled open standard definition on transactions between administrations, citizens and business:

- Non-discrimination and equal treatment: no definition of open standard should precise categorical preferences or refer to specific companies' licensing models since it would protect or eliminate some suppliers or products.
- Ensuring the best technical choice: no definition of open standard should prevent institutions from using the full scale of possible solutions.

It is critical for ETSI to lead the debate to define "Open Standard". In that perspective, ETSI could open a discussion within its Members to quickly adopt a formal definition. But the risk would be to appear as an organisation wishing to close the discussion rather, than the leader of the "Open" debate within the ICT community.

So, instead of insisting on the definition of Open Standards, it appears in the short term that it is of great importance to address the issue of the context and the field of this definition. There already are many different definitions, but none of them are clearly limited to a precise industrial sector or business model.

6 ANNEXES

ANNEX 1: LIST OF FSF APPROVED FREE SOFTWARE LICENSES

GPL-Compatible, Free Software Licenses	GPL-Incompatible, Free Software Licenses	Non-Free Software Licenses
GNU General Public License, or GNU GPL for short.	XFree86 1.1 License	(Original) Artistic License
GNU Lesser General Public License, or GNU LGPL for short.	Affero General Public License	Apple Public Source License (APSL), version 1.x
License of Guile	Arphic Public License	Reciprocal Public License
License of the run-time units of the GNU Ada compiler	The Condor Public License	SGI Free Software License B, version 1.1
X11 License	Original BSD license	Sun Community Source License
Expat License.	OpenSSL license	Old Plan 9 License
Standard ML of New Jersey Copyright License	Academic Free License, version 1.1.	Open Public License
Public Domain	Open Software License, version 1.0	eCos Public License, version 1.1
Cryptix General License	Apache License, Version 1.0	Sun Solaris Source Code (Foundation Release) License, Version 1.1
Modified BSD license	Apache License, Version 1.1	YaST License
License of ZLib	Apache Software License, version 2.0	Aladdin Free Public License
License of the iMatix Standard Function Library	Zope Public License version 1	Scilab license
W3C Software Notice and License	License of xinetd	AT&T Public License
Berkeley Database License (aka the Sleepycat Software Product License)	License of Python 1.6b1 and later versions, through 2.0 and 2.1	Jahia Community Source License
OpenLDAP License, Version 2.7	Old OpenLDAP License, Version 2.3	License of ksh93
License of Python 1.6a2 and earlier versions	IBM Public License, Version 1.0	License of Qmail
License of Python 2.0.1, 2.1.1, and newer versions	Common Public License Version 1.0	The license of PINE
License of Perl	Eclipse Public License Version 1.0	Microsoft's Shared Source License
	Phorum License, Version 2.0	Hacktivismo Enhanced-Source Software License

Clarified Artistic License Zope Public License version 2.0 Intel Open Source License (as published by OSI) License of Netscape Javascript eCos license version 2.0 Eiffel Forum License, version 2 License of Vim, Version 6.1 or later Boost Software License EU DataGrid Software License The license of Ruby	LaTeX Project Public License Mozilla Public License (MPL) Common Development and Distribution License (CDDL) Netizen Open Source License (NOSL), Version 1.0 Interbase Public License, Version 1.0 Sun Public License Nokia Open Source License Netscape Public License (NPL) Jabber Open Source License, Version 1.0 Sun Industry Standards Source License 1.0 Q Public License (QPL), Version 1.0 PHP License, Version 3.0 Zend License, Version 2.0 Vita Nuova Liberal Source License Lucent Public License Version 1.02 (Plan 9 license) Apple Public Source License (APSL), version 2	Agreement (HESSLA) The Squeak license

ANNEX 2: LIST OF OSI APPROVED OPEN SOURCE LICENSES

Academic Free License	MITRE Collaborative Virtual Workspace License (CVW License)
Adaptive Public License	Motosoto License
Apache Software License	Mozilla Public License 1.0 (MPL)
Apache License, 2.0	Mozilla Public License 1.1 (MPL)
Apple Public Source License	NASA Open Source Agreement 1.3
Artistic license	Naumen Public License
Attribution Assurance Licenses	Nethack General Public License
New BSD license	Nokia Open Source License
Computer Associates Trusted Open Source License 1.1	OCLC Research Public License 2.0
Common Development and Distribution License	Open Group Test Suite License
Common Public License 1.0	Open Software License
CUA Office Public License Version 1.0	PHP License
EU DataGrid Software License	Python license (CNRI Python License)
Eclipse Public License	Python Software Foundation License
Educational Community License	Qt Public License (QPL)
Eiffel Forum License	RealNetworks Public Source License V1.0
Eiffel Forum License V2.0	Reciprocal Public License
Entessa Public License	Ricoh Source Code Public License
Fair License	Sleepycat License
Frameworx License	Sun Industry Standards Source License (SISSL)
GNU General Public License (GPL)	Sun Public License
GNU Library or "Lesser" General Public License (LGPL)	Sybase Open Watcom Public License 1.0
Lucent Public License (Plan9)	University of Illinois/NCSA Open Source License
Lucent Public License Version 1.02	Vovida Software License v. 1.0
IBM Public License	W3C License
Intel Open Source License	wxWindows Library License
Historical Permission Notice and Disclaimer	X.Net License
Jabber Open Source License	Zope Public License
MIT license	zlib/libpng license

7 Glossary

API	Application Programming Interface A generic term for the routines, features, and associated rules which allow computer programmers to exchange between applications
Applet	Small application. For example, a Java interactive animation applet could be included within a web page or television enhancement if the user platform includes a Java run-time engine.
Application server	Also called an appserver. A program that handles all application operations between users and an organizations' backend business applications or databases. Application servers are typically used for complex transaction-based applications.
BSD	Updated in 1999, the BSD license is used to distribute BSD software. The owner of the original BSD distribution was the "Regents of the University of California" because BSD originally came from the University of California, Berkeley. The license is very permissive, it authorizes commercial uses, it does not require any copyleft provision and derivative works can even appropriated. This license is very common and there are many examples of BSD code in Microsoft products (like parts of the networking code) or within Mac OS X (using parts of FreeBSD).
Copyleft	Copyleft is a concept drawn by Richard Stallman and the Free Software Foundation as a license that reiterates the four freedoms of the Free Software philosopy to be reproduced in identical terms in the case of redistribution. This avoids the distribution of the modified software restricting the initial rights.
Copyright	A copyright is a set of exclusive rights granted for a limited period of time to protect the particular form, way or manner in which an idea or information is expressed.
Dual Licensing	A software with 2 or more licenses giving to the user to choice one or another license. This choice can be restricted by the licensor depending of the use, the purpose …
FLOSS	Proponents of the Free Software or the Open Source movements often prefer to use the term Free/Libre Open Source Software.
FSF	Free Software Foundation (FSF) is a non-profit organization founded in 1985 by Richard Stallman to support the free software movement (free as in freedom), and in particular the GNU project. The FSF holds the copyrights to most GNU software and some non-GNU Free Software as they require copyright assignment papers from each contributor to GNU packages, so that they can defend the software in court if a dispute arises or if there is a need to change the license of a work.
GNU	GNU is a recursive acronym for "GNU's Not Unix". The GNU project was launched in 1983 by Richard Stallman with the goal of creating a complete free software operating system -- called the GNU system or simply GNU. Its principles are that users are allowed to copy, modify and redistribute it. The GNU project is now carried out by the Free Software Foundation.
GPL	The GNU General Public License (GNU GPL or GPL) is a free software license, originally written by Richard Stallman for the GNU project (a project to create a complete free software operating system). Released in 1991, the version 2 became the most popular licenses for free software.
Intellectual Property	IP is a legal entitlement which sometimes attaches to the expressed form of an idea or of other intangible subject matter. Well known examples of intellectual property include copyrights, patents, trademarks, and trade secrets.
IPR	Intellectual Property Rights are the rights weighing on one's creations:

	patents, copyrights, trademarks, etc.
Java™	High-level programming language developed by Sun Microsystems. Java is a general-purpose programming language with a number of features that make the language well suited for use on the World Wide Web. Small Java applications are called Java applets and can be downloaded from a Web server and run on your computer by a Java-compatible Web browser, such as Netscape Navigator or Microsoft Internet Explorer. For television, a compact version of Java, called Personal Java, has been adopted within the DVB-MHP specification as its standard execution platform for television set-top boxes, and it is more widely used outside the US than within.
LGPL	Released in 1991 and updated in 1999, the GNU Lesser General Public License (LGPL) is a commonly used software license. It is originated from a modified version of the GPL in order to apply to software libraries.
LINUX	Linux is open source software, which is freely available. Due to its stability, Linux has gained popularity with ISPs as the OS for hosting Web servers. Its usage is expected to grow as a server OS as well as for the desktop (see KDE and GNOME). In 1990, Finnish computer science student Linus Torvalds turned Minix, a popular classroom teaching tool, into Linux, which is closer to the real Unix. Torvalds created the kernel, and most of the supporting applications and utilities came from the GNU project of the Free Software Foundation. Many programmers have contributed to the Linux/GNU system.
Middleware	Term used to describe separate products that serve as the glue between two applications. Middleware is sometimes called plumbing because it connects two sides of an application and passes data between them.
Mozilla	Mozilla is a free, cross-platform internet suite whose development was initiated by Netscape Communications Corporation based on the source for their Netscape Communicator.
Open Group	Formed in 1996 as the merger of the Open Software Foundation (OSF) and X/Open organizations, The Open Group is dedicated to promoting open standards and providing certification in a variety of areas, including the Unix operating system and Motif and Common Desktop Environment (CDE) user interfaces. Founded in 1984, X/Open was dedicated to developing specifications and tests for open system compliance. X/Open was involved in unifying the Unix operating system into the Single UNIX Specification. It held the UNIX trademark (upper case letters) on behalf of the industry, which passed to The Open Group. Founded in 1988, the OSF was a coalition of vendors and users that delivered technology innovations in all areas of open systems, including OSF/1, a Mach-based operating system, the Motif GUI and the DCE platform.
Open Source	Free source code of a program, which is made available to the development community at large. The rationale is that a broader group of programmers will ultimately produce a more useful and more bug-free product for everyone, especially because more people will be reviewing the code. Peer review is considered one of the most important safeguards to prevent buggy code, but is often not given enough, if any, attention by software companies. Peer review is a natural byproduct of open source projects.
OSF	Open Software Foundation (See Open Group)
Open Source Initiative	Open Source Initiative (OSI) is a non-profit corporation delivering the OSI Certified Open Source Software certification mark and program. They list software and licenses and they study their compatibility.

Patents	A patent is a set of exclusive rights granted for a limited period of time in exchange for the regulated, public disclosure of certain details of an invention.
Software License	In computing, software that is copyrighted must be licensed to be used. These licenses are primarily written to deal with issues of copyright law and product liability law. They can also concern issues related to patent law, trademarks, trade secret law, and laws pertaining to access to services. Software licenses are generally classified into two categories: proprietary software licenses and free software licenses, depending on their content.
XML	Extensible Markup Language Language developed by the W3C. XML is a pared-down version of SGML, designed especially for Web documents. It allows designers to create their own customized tags, enabling the definition, transmission, validation, and interpretation of data between applications and between organizations. Whether XML eventually supplants HTML as the standard Web formatting specification depends a lot on whether it is supported by future Web browsers. Microsoft Internet Explorer version 5 handles XML, but renders it as CSS, and Mozilla (Netscape) is still experimenting with XML support.