



# Study into the use of Open Source Software in the Public Sector

Part 3

## The Open Source Market Structure

A report directed by

Patrice-Emmanuel Schmitz, Unisys Belgium



## An IDA Study Interchange of Data between Administrations

European Commission, DG Enterprise

**UNISYS**

June 2001

## The IDA programme

IDA is a European Commission driven strategic initiative using advances in information and communication technology to support rapid electronic exchange of information between Member State administrations. The objective is to improve Community decision-making, facilitate operation of the internal market and accelerate policy implementation.

Its [mission](#) is to co-ordinate the establishment of trans-European telematic networks by:

**Promoting** implementation of sectoral networks in priority areas

**Developing** network interoperability measures

**Extending** network benefits to EU industry and citizens

**Co-operating** with Member States authorities and Community services

**Promoting** convergence towards a common telematic interface.

IDA organised a one-day seminar in Brussels on 22 February 2001 to address the use of open source software (OSS) in public administrations. The event brought together around 100 representatives of the Commission, national and local governments and the IT industry. It provided a platform for EU administrations to share experiences, and permitted dialogue with the private sector on the benefits and pitfalls of OSS usage.

IDA also addressed a call for tender related to a "Study into the use of open source software in the public sector" (the present study)

The Study has three components:

**Part 1 The OSS Fact sheet.** An assessment of availability and potential of OSS based solutions, by software category, and a selection of about 100 typical OSS solutions (out of several thousands of OSS "projects").

**Part 2. The report on OSS usage and experiences.** Based on the Fact sheet and a Questionnaire, and on visits in six European countries (France, Spain, Germany, Italy, Belgium, Sweden), the report examines the use/non-use of OSS in the public sector.

**Part 3. The report on market structure and issues related to public procurement.** It describes how OSS may be used / distributed according to their licenses, and how the legal and commercial aspects may impact public procurement objectives, transparency and non-discrimination.

**Unisys Belgium** obtained the contract and provided manpower, project management and support services for the study.

This report has been prepared under the sole responsibility of the contractor. It does not necessarily reflect the view of the Commission, nor does the Commission accept responsibility for the accuracy or completeness of information contained herein



## Modification history

Date	Version	Author	Reason for modification
13.7.01	Draft	P-E Schmitz	Draft
23.7.01	V01	"	Sent
8.8.01	V02	"	Remarks
17.8.01	V03	A. Mennens	Corrections (syntaxe)

### Warning:

The document includes links (URLs) to sites and pages located on the Internet. Since a lot of information - permanently updated – is available there to the public, it would be a non-sense to import all this data verbatim in the present document (it would multiply the number of pages).

The reader of the present document should therefore be connected to the net if he wants to consult these external pages.

The author of the present paper cannot guarantee that all the referred links will stay active and will continue responding to the information need, as they were at the time of writing.



## Table of Contents

Summary .....	7
Abbreviation table .....	9
Introduction .....	10
Methodology .....	12
Collected data consolidation .....	12
The six visited countries and the European Commission .....	12
63 % of interviewees use some type of OSS .....	14
Only 8% of public sector servers includes OSS components ....	14
Reasons why OSS is used in Public sector .....	15
Interoperability and respect of standards.....	15
Security and availability of the source code.....	16
Functionalities, Quality and ease of use.....	16
Cost reasons, Stability, Training and Support.....	16
Human factors.....	17
Political reasons and Independence from the vendor.....	17
Other reasons.....	17
Why is OSS not used in Public sector? .....	17
Weight and quality of existing ICT infrastructures.....	17
Contractual and legal engagements.....	18
Fears of global budget reductions.....	18
Lack of pre-installed system delivery.....	18
Lack of accountability.....	19
Interoperability problems (concerning hardware) .....	20
Interoperability problems (concerning software) .....	20
Reduced set of public sector oriented applications.....	20
Reduced set of home applications.....	20
Lack of public sector "turn-key" distribution.....	21
OSS Market evolution .....	22
Analyst estimations on Server market .....	22
The battle of the next two years .....	23
Plug-and play Servers.....	24
Respectability, Snowball effect and politics.....	24



Growing need for Application Service Providers (ASP's) .....	26
The move to Mainframes .....	26
Desktop domination .....	27
The All or Nothing Change on desktop .....	29
Service market .....	29
Service provider groups .....	29
Categories of Support .....	31
Market size and Trends .....	33
OSS Model evolution .....	33
From freedom to standard .....	34
Development model .....	34
The projects / the project developers .....	35
A multiplication of Alliances to support OSS .....	35
The Open Standards .....	36
The OSS organisations .....	37
Technical co-existence Proprietary/OSS .....	37
OSS and Innovation .....	38
The Business model .....	40
The OSS TCO analysis .....	42
TCO notion .....	42
GVC Government value creation .....	43
Benchmarking with Best Practice .....	46
Is general TCO estimation possible "in the labs"? .....	48
Who should estimate? .....	52
Government action .....	53
Direct support to OSS? .....	53
Support to interoperability .....	54
Standards and best practices: .....	55
Grant interoperable public document access .....	57
Grant interoperable Internet services .....	57
Legal Issues .....	58
A matter of license .....	58
A world of diversity .....	60
Evolution toward hybrid or dual licenses .....	61
Legal coexistence OSS/Proprietary .....	62



The specific GPL controversy.....	62
The Liability due to OSS delivery .....	65
Principles.....	65
Validity of the license agreement.....	66
Gift or for sale - Virus liability.....	66
Distribution by whom? To whom?.....	66
Market and fair competition .....	67
§ 81 of the EC Treaty.....	67
§ 82 of EEC treaty.....	69
§ 87 of EEC treaty.....	70
The Software Patent question .....	70
The Law.....	71
The reality.....	72
The Controversy.....	73
The EPO demand.....	74
The reactions.....	75
Recommendations .....	77
Information Collection.....	77
Reflection on OSS benefit of ownership.....	77
Respect of standards.....	77
Shared investment in new standards.....	77
Certification authorities.....	78
Call for tender.....	78
Promotion of a global transparent market.....	78
A clear limit to patentability.....	78

# Summary

With the exception of education, Open Source Software (OSS) is still not extensively used in most of the European Member States' public administrations: OSS advocates opened the way, and in some leading countries they are now relayed (or recuperated) by political responsible ministries, but concrete large scale OSS projects are still rare.

After examining the market we have to conclude that on general-purpose servers as well as on office desktop, Open Source software will present tomorrow the most realistic, and sometimes the only real technical and economical alternative to Microsoft products. This is due to a growing support of many Information Technology and Service giants (following the example of IBM), as distributors and specialised "Linux SMEs".

Open source will never "dominate" proprietary software because it is not the property of anybody. The natural tendency of the software industry is to continue to make money out of innovation. New proprietary software will try to gain its market share every day, and Microsoft in particular continues to add innovation, quality and functionalities to its products, in order to progress in the high-end industry leaders' office automation and global integration (translation, voice, people and mobile users management, knowledge, data, document and process management market).

At the same time, despite all new versions and improvements to their products, the proprietary vendors will never win against Open Source or dominate it. There are two reasons for this: they cannot buy the open source movement, nor can they delete it by de-motivating the thousands of Open Source advocates. The paradox is that the stronger Microsoft is and the more it tries to extend its domination, the stronger the reaction and motivation of the OSS community will be. In addition, some proprietary products that will be unable to triumph on the market despite their qualities, will turn to Open Source to survive. In a way, Microsoft itself generates Open Source as an "antithesis", and will then be obliged to look for growth in other popular (games, culture) or high-end sectors.

Up to now often reactive, the open source community and the performing development model issued from the Internet will also become more innovative: in 6 months, from December 2000 to June 2001, the number of projects declared at host sites like Source Forge grew almost 100% (from 10.000 to 20.000). A lot of them will die. A number of them will become tomorrow's standards.

On the service and integration market, a new economic model appears: from licensing to services. It was adopted as well by industry majors (IBM and followers) as by small and medium enterprises or start-ups.

On the server market, Open Source is already used on a large scale. It fully responds to the needs concerning several functionalities and as such it provides the best value for money: web servers, dedicated plug and play servers, database and ASP servers.

The NT / Windows 2000 market will continue to grow in size and in value on high-end servers and performing business applications. This is due to the quality of proprietary developments made for these platforms and their "de facto" standardisation (attracting most of the commercial developers).



On desktop and for general office tasks, the number of variants, distributions, more or less mature OSS projects and the lack of interoperability between documents formats will continue to discourage non-hobbyist individuals and many SMEs. They prefer to pay more for simplicity and conformity to the dominant market standard. But at the same time, large administrations and some large desktop consuming enterprises will be more and more sensitive to cutting license cost where a significant scaling effect will be possible (selecting and adapting an OSS distribution to the needs, in order to make installations generic and to simplify users' management).

OSS servers can be introduced progressively on LANs and on the Internet without anybody noticing it. The desktop migration to OSS however is a strategic decision and an "all or nothing choice": thousands of users will migrate at the same time and a lot of existing office documents have to "migrate". Before reaching this decisive point, OSS office suites will have to improve their interoperability at document exchange level, thanks to fully compatible document formats, based on a set of XML data structure (adapted to interchange of data between administrations.)

In order to realise this portfolio of common XML data structures, governments and public sector agencies should not "reinvent the wheel" on a national basis with the risk of re-constructing e-borders. Rather, they should adopt in a "peer to peer" relation the most significant advantage of the Open Source movement: *the development model*.

The role of the European authorities may be to provide the resources for a common mediator or facilitation Internet host (a kind of XML data structure Source Forge) that should impose the respect of the same set of standards (in a neutral way, both for OSS and proprietary software) and respect the diversity of languages and cultures: all government or public sector agencies should be invited to contribute with their findings and innovations.



# Abbreviation table

AIPA	Autorità per l'Informatica nella Pubblica Amministrazione (Italy)
API	Application programming Interface
ASP	Application Service Provider (software providing centralised applications to web clients)
BMWi	Bundes Ministerium für Wirtschaft und Technologie (Germany)
BSD	Berkeley Software Distribution
DG	Directorate General (of the EU Commission)
EU	European Union
FSF	Free Software Foundation (of Richard Stallman)
FUD	Fear – Uncertainty – Doubt (an effect that makes IT managers and users hesitant at the time of taking strategic decisions)
GNU	Gnu's Not Unix (general project of the FSF)
GPL	General Public License (of the FSF)
GVC	Government Value Creation
HTTP	Hyper Text Transfer Protocol (of IETF / W3C)
ICT	Information and Communication Technology
IDA	Interchange of Data between Administrations (EU programme)
IDC	International Data Corporation
IETF	Internet Engineering Task Force (Standardisation group)
IP	Internet Protocol
ISPO	Information Society Project Office (EU Comm.)
KBSt	Koordinierungs- und Beratungsstelle (Germany)
LAN	Local Area Network
MAP	Ministerio de las Administraciones Publicas (Spain)
MTIC	Mission interministérielle de support aux Technologies de l'Information et Télécommunication (France)
MS	Microsoft
OS	Operating System (of a computer)
OSI	Open Source Initiative
OSS	Open source software (= Free Software; = Libre software)
PAGSI	Plan d'Action Gouvernemental pour la Société de l'Information (France)
PC	Personal Computer
RAM	Random Access Memory (of a computer)
SME	Small and Medium Enterprise
SQL	Structured Query Language
W3C	The World Wide Web Consortium (Standardisation group)
SME	Small and Medium Enterprises
SMP	Symmetrical Multi Processors
TBO	Total Benefit of Ownership
TCO	Total Cost of Ownership
USB	Universal Serial Bus

# Introduction

At a time when one vendor increasingly dominates the PC industry (desktop operating system and office suite) and a growing part of the server industry, it is surprising to discover that its strongest challenger is not a commercial rival but a motley collection of free software tools and operating systems called "Open Source Software".

The power of this movement is multiple: nobody can buy it, and it is supported by thousands of enthusiasts that cannot be discouraged by anything. The quality and rapidity of OSS development is unanimously recognized as "amazing", even by proprietary vendors. A new dynamic is created by the fact that the source code of such software can easily be studied by other programmers and improved, the only condition being that such improvements must also be revealed publicly and distributed freely in a process that encourages continual innovation.

From an operating system called Linux, named after a student from Finland who wrote its core code, to a web server named Apache, put together as literally "a patchy" set of updates to older software by a band of volunteer programmers, these open source programs are emerging not just as inexpensive but also as more robust and dynamic alternatives to commercial software.

Like in the private sector, most public sector departments use open source software if they find in it a response to their needs. The first step or entrance ticket for open source software into administration has not been easy. Until a couple of years ago, Linux and most other open source software were just ignored in most infrastructures, forcing some programmers and system administrators to slip it into the organization through the backdoor. Today, many of the same organizations are discovering that open source software may provide some realistic solutions into their organizations.

While this phenomenon surprises some analysts, it should not surprise those with some sense of history. Open source software, largely funded by the (United State) government, was the wellspring of the creation of the whole computer industry. To this day it still lies at the heart of how the Internet came into being. For decades some governments and public institutions in education had stimulated open source software and open standards through a combination of key funding agencies, administrative oversight of software standards and government purchasing rules. While such software never totally disappeared, its prominence was undermined by the privatisation of the Internet and the commercialisation of areas of software once dominated by open source options. Largely, this was due to the fact that in the early 1990s, the US federal government pulled back from its commitment to open standards and support for open source software. This left the way open for increases in proprietary, incompatible software and attempts to dominate the computing world with its own proprietary standards.

If open source software is re-emerging today as an important force, all opinions and literature demonstrate that it is largely as a reaction from the "small people" against one specific company: Microsoft. Competitors who have seen their own proprietary alternatives sink under the Microsoft steamroller have suddenly seen alliances with open source software as a chance to halt the Windows monopoly. As a matter of fact, the opportunism of these alliances have created a whole set of tensions during the last two years, and we do believe that this kind of interested alliance is unlikely to make open source software a real neutral and objective alternative to proprietary software.



Considering all these facts and sometimes contradictory tendencies, the present report will look at the OSS market structure: the use, the forecasting concerning servers and desktop, as well as the possible issues.



# Methodology

This report forecasts and sizes the use of OSS in the public sector market. Three axes are examined:

- Server market
- Desktop market
- Service and support market

The report is based on:

- 66 Interviews and questionnaires with public sector and EU representatives and Internet inquiries concerning public sector usage in the six selected countries (France, Spain, Germany, Italy, Belgium, Sweden).
- Visits in the studied countries.
- Participation to seminars.
- six months' technological survey (January – June 2001).

## Collected data consolidation

### The six visited countries and the European Commission

The second part of the IDA Study into the use of Open Source Software (OSS) in the public sector demonstrated a clear difference between the countries visited.

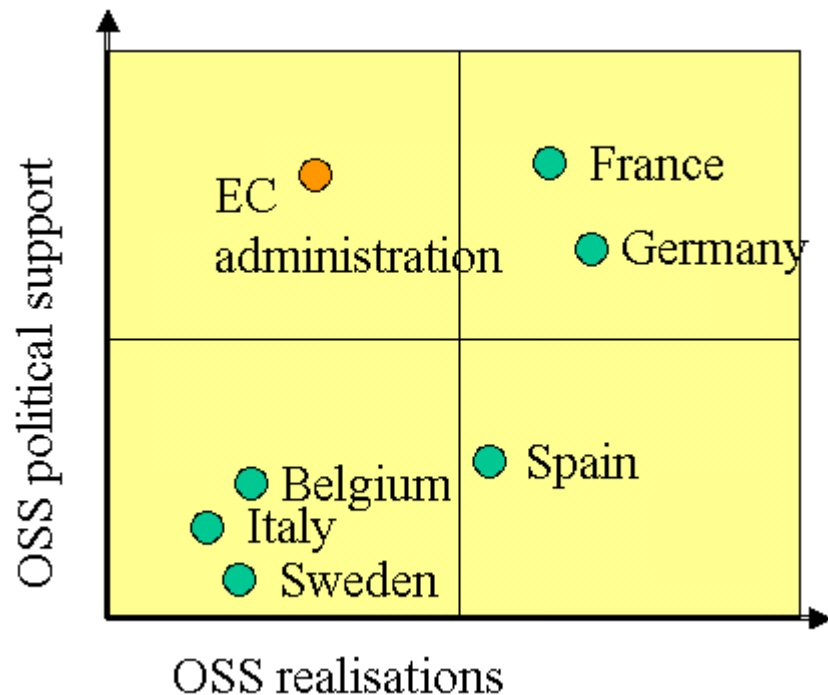
France and Germany are the two innovative leaders in the field: Germany for more concrete realisations and guidelines, France for a growing government support to open standards and open source.

Spain is an active follower concerning specific departments where competent open source advocates have demonstrated the efficiency, the best value for money and the supportability of the solution, and where the installation of standard OSS distribution can provide a scaling effect. However, the official political support to open standards and open source is still limited.

The European Commission administration is in an opposite situation: the political support is high with the e-Europe initiative, the position of the DG Enterprise, the inclusion of open source projects in the Information Society IST 5<sup>th</sup> research and development programme, the IDA programme etc., but the internal rate of realisation is still quite low (with some exceptions in external or marginal organisations like EIONET). This will change gradually as some OSS will be included into the list of supported software in the course of 2001.

In Belgium, Italy and Sweden, the existing realisations result from individual efforts that are still limited in volume and are not – until now – actively supported by a government policy.





There is a clear division between the server and the workstation markets.

There are several categories of servers: pure Web servers (where Apache has already 61% of the global market), dedicated file servers, and the ten to twenty times more important (in value) market of the general-purpose servers (on LANs, Intranets). This is now the main area of OSS expansion, with mature solutions for file servers, database and application service providers.

Globally (general-purpose server and web servers), the percentage of use of OSS on public sector servers is still estimated relatively low (8%). As OSS may be embedded in various dedicated devices, this absolute percentage does not reflect real full open source integrations. These are still limited in number (or even non existing outside the education sector in Sweden and Italy) but may grow fast as soon the IT directorates will start to apply government recommendations for more neutrality and open standards (e.g. in France). In a first stage, the growth will mainly be concentrated in replacement or extension of proprietary Unix solutions.

It will also mainly concern “Linux utility servers”, which are the file, print, e-mail, and Web servers most commonly used for Internet infrastructure. As Linux grew up in this utility server space, it was frequently outside of the production data center environment. The very benefits of the Linux kernel, low initial cost and robustness, generally resulted in fewer software support contracts compared to equivalent Unix and Windows utility servers, within relatively simple infrastructures where it is proven that OSS can provide an equivalent panel of solutions, or for new installations. In complex old infrastructures with thousands of clients, where many types of functionalities, hardware and drivers have taken years to be well integrated or when sophisticated client-server applications have already been written for specific client types, the implementation of OSS may be perceived as a “new – unwanted” risk by IT managers and will depend more on a “political” pressure. On new or re-engineered infrastructures, Linux will grow up in mission-critical applications, thanks to the new generation OSS ASPs space.

On workstations, with the exception of some education organisms, the use of an OSS operating system (GNU/Linux, FreeBSD) and of an open office suite for example, is even more limited (not more than 1% actually, with an exception for the education sector).

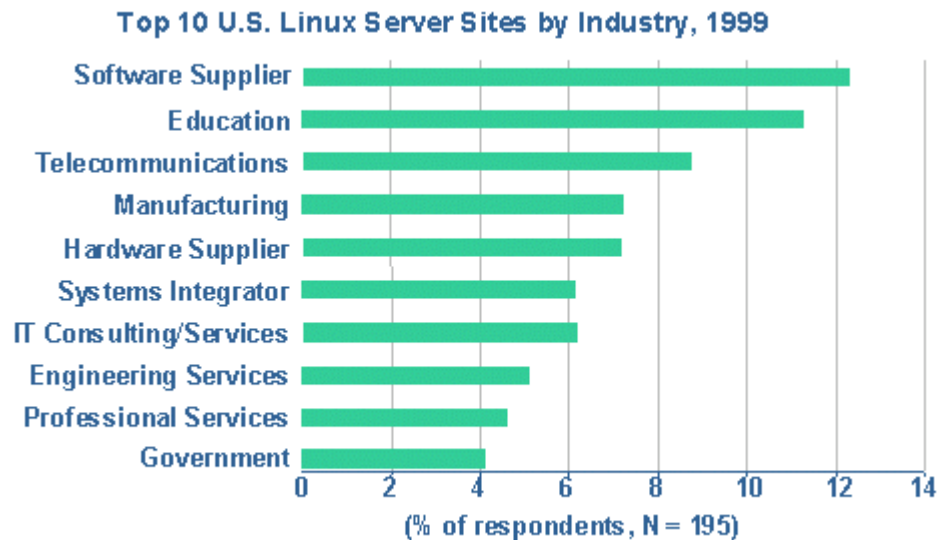
## 63 % of interviewees use some type of OSS

Even if 63% of the Public sector IT managers declare using at least one OSS component, this does not represent a great volume, in value or even in units, on their network. Most of the installations are limited to a dedicated web or file server.

This 63% percentage may be compared with the 56% obtained by Forrester in August 2000<sup>1</sup> within the top 2500 US companies.

## Only 8% of public sector servers includes OSS components

From those answering positively to the question “Who uses Linux servers?” IDC estimated<sup>2</sup> for the year 1999 that in the US, only 4% of its respondents belong to the government sector (government was here separated from education):



Confirming the findings of IDC in the US regarding Linux sites by industry in 1999, the European public sector in 2001, with the exception of education, is not a major OSS user either, with about 8% (but still twice the IDC 1999 finding).

In many administrations however, the use of some type of OSS is still experimental. The greatest part of budgets are still used as before, based on long term prevision and multi-annual frame contracts, to support proprietary infrastructure.

<sup>1</sup> Open Source Cracks The Code – by Carl D. Howe – August 2000 Forrester Report

<sup>2</sup> “Linux server gains shares”, - IDC IT forecaster April 11, 2000



A reason for that is the “weight of the past”. At a moment when specialized consultants, University professors or Open source advocates are already several steps ahead, Public Sector IT managers, who have several hundreds of servers and thousands of workstations to manage, are still implementing strategies that were defined four to five years ago. Their first priority then was to escape from hardware constructors’ proprietary platform dominations. In this context, environments like Unix and NT were perceived as the best value for money.

## Reasons why OSS is used in Public sector

The following table presents the results of the interviews and questionnaires (see part 2 of the Study). The reasons to use OSS in the public sector (for those who are using them) are presented in decreasing order of importance (10 being the maximum):



### Interoperability and respect of standards

With more than 80% attributed to these criteria, the interoperability and respect of standards of the OSS are the main reasons for using OSS in the Public Sector. The main strength of open source software is that it is “constructed for interoperability” and “closely associated to open standards”. OSS is considered to better respect standards because no proprietary standards are used to “protect” the vendor captive market and it is in the interest of everybody to achieve the best inter-operability. The permanent research for public common standards makes OSS more convenient for long-term interoperability. A threat against standardisation may come from software patents if they prevent developers from using common “commercial” standards. The problem regarding interoperability is also the main reason “not to select” an open source solution, if it appears to be unable to integrate proprietary documents and file formats (and this is mainly the case for desktop office applications).

## Security and availability of the source code

The availability of the source code and the right to modify is also a very important factor. This is not only because of a real intention to modify the software, but also because OSS contains less “black boxes”. Understanding how the system works is a cornerstone of public sector requirements in terms of transparency. No software (OSS or other) will probably ever be 100% secure, but at least with OSS you will have no (or less) backdoor(s), no electronic spy that may be totally hidden somewhere in the software.

There is obviously no such thing as an infallible encryption system. As far as electronic communications are concerned, experts believe<sup>3</sup> that open source software offers better protection against backdoor entry and could partially meet the need for on-line security. In other words, the content of open source software is fully verifiable whereas with 'closed' systems the possibility of activation from a distance could not be ruled out.

## Functionalities, Quality and ease of use

OSS is generally of better software quality and a higher reliability is obtained in many cases. It is true that no one grants that a specific OSS development will ever be done. Once it is made however, it has usually been tested and commented by many “free-minded” developers. There is no “time pressure” on the development, as is the case with vendors promising their release for a specific date and then providing bugged versions just to respect the planning engagement.

## Cost reasons, Stability, Training and Support

The first perceived advantage of open source models is the fact that open source software is often made available gratis or at a low purchase cost. Acquisition costs, however, are just one component (generally about 20%) of the Total Cost of Ownership (TCO). Users have to consider carefully other costs like deployment, data migration, training, support and interoperability. The solution will never be “for free” and investments for OSS may even be more important in terms of training / implementation. The License cost cutting however, may become very attractive depending on the number of users (scale effect), and depending on the presence of a previous existing Unix environment: Since Linux, BSD etc. technically operate as Unix does, it reduces the need for training within the IT team if for example Gnu/Linux replaces a proprietary Unix.

Stability is due to the fact that no proprietary vendor will impose a migration to the user, just because an old hardware platform is not “supported anymore”, forcing the user to purchase a “new version for the new platform” of the only slightly adapted old product. This is another important component of the cost, which becomes sensitive at a time when proprietary vendors (Oracle, Microsoft) undertake migrations or re-evaluate the price of their licenses.

---

<sup>3</sup> Bart PRENEEL, professor of cryptology at the Catholic University of Louvain, - reporting to the European Parliament during the “Echelon” espionage system discussions 16 October 2000.



## Human factors

Compared to the use of proprietary “closed” software (even proprietary Unix), the impact of the Open Source model on the human factor may also be significant. From “passive” licensee (with no real power on the tool’s quality), the development team will feel promoted to “member of a community of peers”, contributing to forums, exchanging codes and tips. This may also produce a real “waste of time”, but this time may be transformed into a higher motivation and more competence and expertise.

## Political reasons and Independence from the vendor

Apart from a small percentage of “real OSS advocates”, the ‘political’ reason to prefer OSS is generally not ranked first. However, it should not be under-valued. During interviews and conversations, the question of the “dominant vendor” arises automatically, with a curious mixture of “Love-hate” effect. Most IT managers consider at the same time with respect the Microsoft evolutions and adaptability from NT, W2000 to XP, the XML standard, .Net, while they require alternatives and independence from any private vendor (this is not specifically directed against Microsoft) to implement large-scale e-government practices.

## Other reasons

With some exceptions, the normal Public sector IT manager wants to use OSS in the same way as he used proprietary solutions. The availability of the source code is important as a “guarantee” that the software can be taken over and that it contains less obscurity, black boxes or backdoors. Few public sector IT managers are really interested in making their own version / modification and to redistribute it to third parties. Free internal diffusion allows them to develop prototypes of distributions (a dedicate selection of software for their servers or desktop) and to redistribute it without taking licensing problems in consideration.

## Why is OSS not used in Public sector?

### Weight and quality of existing ICT infrastructures

The European Commission provides a good example of an existing large ICT infrastructure of high quality, which cannot easily be reoriented to OSS. The Informatics Directorate has done substantial work to gain its independence from proprietary constructor’s infrastructures and moved to “standard” Unix and NT infrastructures. Finally a high level of internal standardization on application servers and on desktop has been reached with the MS/Windows Office suite. Therefore they are reluctant to reintroduce heterogeneous components, which they will be obliged to support, with all the related interoperability and data migration problems that may occur.

Comment:



This is a normal situation, due to the respect of the existing users, data and applications. IT managers must avoid making a permanent clean sweep of the past to embrace the last technology. However, they are gradually aware of the necessity for neutrality and political diversity. This leads them to open their requests for proposals to a wider range of products, giving as much importance to certified interoperability based on open standards as to pure functionalities or technology.

## **Contractual and legal engagements**

Another reason why OSS is not (yet) used in public sector is that their procurement of IT infrastructure is often based on long-term frame contracts, with traditional large IT partners. Few of these partners have already integrated the new OSS proposal, at the level of their support and technical teams. Service level agreements based on existing infrastructures are often contracted for periods of 5 years and more.

Comment:

There is no immediate solution to this situation since existing contracts need to be respected. In the new contracts and calls for tenders however, respect of open standards could be included. Upcoming calls for tender provide the opportunity to give OSS a chance. OSS may be introduced when launching new e-government services or when renewing previous Unix support agreements.

## **Fears of global budget reductions**

As a Belgian Army IT manager admitted, the problem is not to obtain enough money for the purchase of ICT components and licenses, but rather to find the means to develop the human resources component. These are part of the global ICT budget needed. It is the main fear of IT managers that their resources for the coming years will be globally reduced if price reductions are obtained for licenses, or if they do not spend the software budget for licensing as it is foreseen. The Public sector IT budget models are currently not adapted to the use of open source software.

Comment:

More consideration should be given to the OSS Total Cost of Ownership (TCO) model, thus enabling budget transfers (from licenses to services and training) according to possible use of free or low-cost licenses. The efforts of IT managers to obtain a better value for public money should be taken into consideration, compensating the lower cost of licenses with higher services and training budgets.

## **Lack of pre-installed system delivery**

At first sight, the impact of the lack of a pre-installed system delivery with OSS, would seem bigger on individual end-users than on a large public sector organisation. In reality however, people like to use the same tools at work as they use at home. The fact that public sector high-level managers often have considered OSS as “hobbyist” solutions, is also due to the fact that OSS systems are not standard installed features “on every system at the shop”. As the Free Software Foundation admitted<sup>4</sup>, it was hard to find companies that currently sell general-purpose computers with GNU/Linux preinstalled. The situation changes related to servers, with more constructors integrating the Linux kernel alternative as a standard system (quite often in embedded systems or on dedicated servers) but with non-free software added.

The end-user wanting to purchase a desktop will probably have no choice (with the exception of Macintosh). Moreover, the desktop hardware and the mandatory pre-installed operating system are bundled together, without any possibility to change or to ask a price reduction if the box is delivered without software<sup>5</sup>. End-desktop users wanting to install Linux have ordinarily no other choice than to create a separate Linux partition, or to re-initialise the hard disk to install Linux from scratch. Of course, the end user will have to resolve all kind of installation, drivers support and interoperability problems. Worse, the vendor will probably refuse any guarantee or support service – even in case of hardware failure - as long as the original environment is not restored.

Comment:

The software price (Operating System and applications) should be more transparent for the end-user and separated from the hardware price, reflecting the real cost. The end-user should have the possibility to purchase the hardware separately. “Forced sales” are illegal and this should be applied also to software.

## Lack of accountability

The lack of accountability is losing importance as a reason for not using OSS in strategic applications. Annual support and maintenance contracts are now provided both by distributors, hardware vendors and integrators.

The availability of such support contract, not only for the Operating System, but also for any used application is vital, because it is not the mission of the average public sector user to provide the last level of support for IT applications. Even if they ask for and receive the source code, most of them will use the solution “as is”. As is the case for an efficient commercial enterprise, a public sector department has to concentrate on its “core business” and use a reliable external partner when an occasional specialised problem occurs. Without any kind of support contract (e.g. if the software is downloaded for free, “as a gift”) the user will probably still receive efficient (although unpredictable) free support when he launches a open demand in an internet user group. The support will come from everywhere in the Internet world, but no one will accept any liability for this service.

---

<sup>4</sup> <http://www.gnu.org/links/companies.html>

<sup>5</sup> In the US, The Washington DC appeals court ruled (end of June 2001 – Microsoft case) that simple product bundling – in which one product is sold or integrated with another – does not violate the law in high tech (the decision concerns software integration – this is to evaluate case by case).

## **Interoperability problems (concerning hardware)**

Potential interoperability problems for the hardware installed are a reason for the public sector not to work with OSS. Printers, scanners, video cards and other hardware require drivers to be supported (recognised by the operating system). It may take (much) longer for OSS systems to obtain these drivers and support.

## **Interoperability problems (concerning software)**

Another reason is the potential interoperability problems related to software. Exchanging documents between different applications requires conversions (from one file format to another). To do this, applications like spreadsheets or word processors include 'filters' to "import from..." or to "save as...". Proprietary applications do not publish the totality of their proprietary file format. When documents are complex (including table of contents, graphics, macro instructions and programs) filters are often unable to convert them without losing some data or some part of the presentation.

The lack of performance of filters to and from the proprietary products is a major limitation to OSS implementation in office environments, where interchange of data is important. Ideally, interoperability should be measured on the basis of open standards (importing and exporting word processing, spreadsheets, presentations, projects etc. from and to standard interchange formats). In the reality however, users will measure interoperability by importing/exporting documents directly from and to the dominant Microsoft Office suite. This "de facto" situation is a handicap for all tested Open Source office solutions, that were never able to achieve 100% interoperability as soon as a complex structure or macro are used (as in elaborated spreadsheets).

## **Reduced set of public sector oriented applications.**

The number of available commercial applications that can run on OSS Operating Systems is relatively small compared to those available for Windows and for the proprietary Unix. By nature the cooperative development model of OSS answers to collective needs: mass projects where independence and transparency are needed (operating systems, networking, graphical/image editor etc) but it will not concentrate efforts on a specialized application, requiring expert business competences (like accounting, bookkeeping, project management, complex workflow or workgroup management or other branch software). This is more a limitation than a risk, and indicates that there are different domains where proprietary and free solutions may co-exist.

## **Reduced set of home applications**

A number of enterprises have come to use Open Source Software thanks to the individual motivation of OSS advocate employees. To progress, and crossover from the server market to the desktop market, Open Source Solutions should also be attractive at home. As long as the panel of games will remain reduced and as long as, for example no large encyclopedia will be able to seduce families, the home OSS market will stay focused on hobbyists.

## Lack of public sector “turn-key” distribution

Excessive fragmentation is a logical consequence of the open source “business” model (more than 150 different distributions). The various releases are confusing and it takes time and expertise for IT managers to proceed to an evaluation of the 5 to 6 main distributions and to decide which one – if any – and which part of it, will correspond to their needs, may integrate in and will be supported by their existing IT infrastructure. Without specialised consultants or system architects, it may be difficult for users and even IT managers to extract the optimal configuration from the multiple OSS projects (about 10.000), with very different levels of maturity, each of them claiming to be the best or promising smiling futures. Without reference to the famous “Windows-Linux” contest, we have to consider the longevity of systems like IBM OS/400 facing the Unix, partly because it offered a single proprietary road map for most of the needs.



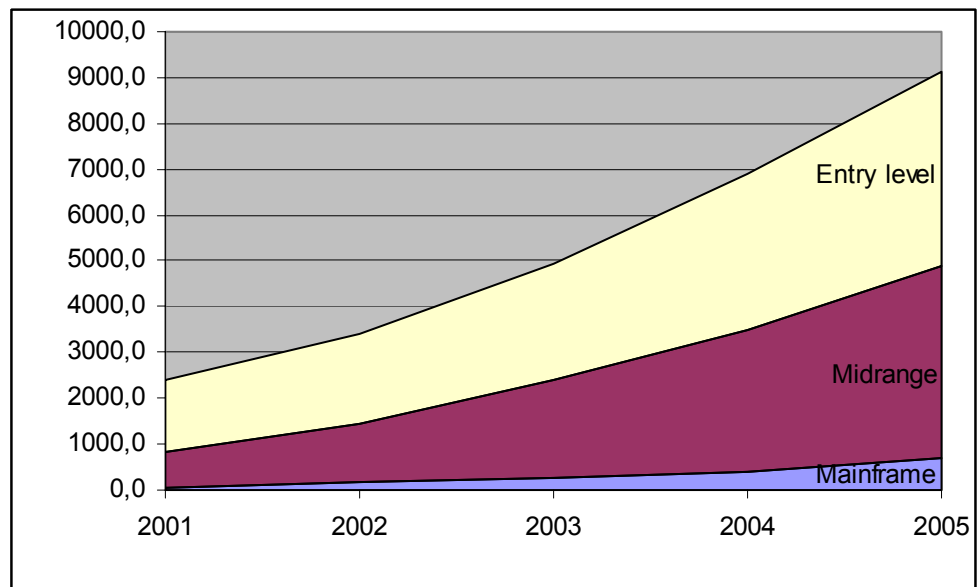
# OSS Market evolution

## Analyst estimations on Server market

Each analyst has developed its own estimation about the OSS market growth, and these estimations vary strongly between, for example, Gartner and IDC, Forrester, the Meta Group or Giga. On one point, all analysts agree: in the coming years, the operating system duopoly (Linux and Windows NT/2000/XP) will continue to grow and to reduce the proprietary Unix, OS/400, Novell etc. market share.

According to the economic slow-down in 2000-2001, we estimate that the most prudent estimation (from Gartner) should receive preference. Starting from a reduced share of the global server market (10% worldwide shipments at the end of 2001), the yearly forecasted growth for Linux on five years is as high as 40% per year, both in units as in revenue (as illustrated below).

Illustration: Linux server market revenues, in Million \$



For the Linux server market, Gartner forecasts an average growth of nearly 40% per year from 2001 to 2005<sup>6</sup> in revenue: from 2.422,1 to 9.142,5 million \$

	2001	2002	2003	2004	2005
Mainframe and super-computers	35,4	159,9	281,1	410,9	708,7
Midrange, from 10.000\$ to 100.000\$	779,4	1.270,0	2.128,6	3.070,4	4.161,3
Entry level from 0 to 10.000\$	1.607,3	1.997,3	2.523,5	3.408,3	4.272,5
<b>Total</b>	<b>2.422,1</b>	<b>3.427,2</b>	<b>4.933,2</b>	<b>6.889,6</b>	<b>9.142,5</b>

<sup>6</sup> A end-User analysis: "Linux Server Market share: Where will it be in 2001 and how will it grow?" By Jeffrey J. Hewitt – published May 30, 2001



Forrester<sup>7</sup> investigated in the top 2.500 companies and expected the real progression in 2003 and 2004 where, globally, open source will displace 20% of licensing dollars (from licensing to services).

The META Group analysed<sup>8</sup> the progression of Linux as a management system and forecasts a strong growth of Linux on server platforms because of its flexibility, low cost and suitability for embedded systems. It estimates however that the same global 2500 companies will prefer Windows 2000 and its descendent for business management platforms by 2004.

The mentioned Gartner Dataquest report<sup>9</sup>, forecasts 10% of the server market for Linux at the end of 2001, based on phone calls to 724 U.S.-based respondents (small and larger organizations as well as educational institutions, Internet service providers and application service providers).

Contrary to Gartner, International Data Corporation (IDC), which provides regular surveys about the growth of Linux, estimates that Linux has already 27 % of the same server market in mid-2001. It is still behind Windows which holds 41%. IDC obtains the higher Linux percentage by considering post-delivery installations, downloading or replicating the same Linux distribution on many machines and giving a second "server" life to old PCs.

## The battle of the next two years

Without any doubt, the real battle will be on the server market for the years 2002-2003.

After Linux was reported by IDC to be the fastest-growing Operating System server environment during the 1999/2000 period with a 132% growth, this slowed down in 2001. Analysts like Gartner and Meta estimate that Windows 2000, XP and its descendents, by providing a more compelling platform than its NT predecessor, will dominate large areas of the server market in 2004, such as the management platform market<sup>10</sup>.

Although IDC estimates that Linux is currently taking more of the market (25% to 27% in number) Gartner estimates that Linux runs on nearly 9 percent of U.S. servers, with worldwide projected sales of nearly \$2.5 billion, reaching about \$9 billion in 2005.

The real difficulty to estimate the server market is that there is no clear border between the server operating environment (which includes also PCs and workstations running server software and configured to act as a server) and the real departmental servers.

---

<sup>7</sup> The Forrester report, "Open source cracks the code" by Carl D. Howe – August 2000

<sup>8</sup> Service Management Strategies – 15 May 2001 – [www.metagroup.com](http://www.metagroup.com)

<sup>9</sup> A end-User analysis: "Linux Server Market share: Where will it be in 2001 and how will it grow?" By Jeffrey J. Hewitt – published May 30, 2001

<sup>10</sup> The Meta group – Service management strategies – by Glenn O'Donnel, 15 May 2001



## Plug-and play Servers

These dedicated servers with embedded Linux typically associate an Intel (or compatible, AMD etc.) low cost processor, and a GNU/Linux distribution reduced to a dedicated usage. They are low-cost, stable and reliable, responding exactly to a specific need, but providing little scalability (although it can be provided through clustering, just by linking servers and providing outstanding performances, as the Google search engine demonstrates).

A new generation of low-cost simplified servers is coming on the market. Designed for small administrations and enterprises, they will be presented as "out of the box" plug and play servers with an extremely simplified installation (just enter the IP address). These dedicated servers will be based on a Linux distribution, Apache (web servers) or SamBA (file servers) for a global price between 1000 and 2000 euro<sup>11</sup>.

Our opinion is that from now on to 2004, the low-end dedicated Linux servers will grow up faster, to represent 30 to 40% of this market in units.

## Respectability, Snowball effect and politics

Contrary to the embedded systems, a generalized OSS deployment on classic, general-purpose or high-end servers is a major architecture decision point which must be discussed in the frame of the overall ICT strategy: stay on commercial Unix or NT, adopt Linux (or another OSS system) or embrace the new MS Windows 2000 universe.

In the public sector, while Microsoft is dominant on desktop with Windows, it is still rather a challenger in the large-scale server market, where the presence of proprietary Unix is still strong.

For Public sector IT managers (and that is also true for the EU), implementing a Microsoft server solution is still much more cost-effective and strategically appealing than to continue with Solaris or other high-end Unix or Mainframe solutions. The market share of Microsoft is therefore growing.

Today, Linux represents only a very small part of the general purpose enterprise or public sector network servers, and its success on dedicated servers will not necessarily translate into success on the management servers side.

---

<sup>11</sup> 90% of these new dedicated servers are Intel/Linux, as the 3Com internet server and Superstack 3 webcache, the Right Vision Eye Box One and Eye Box Pro, the Sun Cobalt CacheraQ4, RaQ4, Staqware...



Although the position (and growth) of Windows on servers is not in immediate danger, the impact of Linux is more serious for the future Microsoft .Net initiative which is widely based on server market control. Microsoft .Net and its authentication component HailStorm make use of XML to pass information between (multiple Operating Systems) computers, however many .Net components-such the Passport, server-based software as SQL Server and BizTalk e-commerce server run only on Windows. Therefore, Microsoft needs to keep control of the server operating-system market if HailStorm and all the .Net services and subscriptions associated with it are to succeed.

Taking decisions about supporting GNU/Linux distributions on the general purpose public sector and e-government servers (in addition to NT and Unix) takes longer because of the weight of existing infrastructure (variety of materials, long-term frame contracts, complex business applications - some of them with "fat" client-server design).

Starting from now, several factors will gradually equilibrate the market in favour of Linux:

- Growing support and promotion by large constructors.  
Linux will be more presented as the supported "Unix standard – reference" environment. As Linux is now supported by several major server providers as HP, SGI and Dell on Intel models, as Compaq (also on Alpha models) and IBM (on S/390 zSeries and on AS/400 iSeries), it provides now an image of security, universality from desktop to main frame and long-term stability for administrations and enterprises.  
In Public sector, Linux will mainly attack the previous proprietary UNIX market and will be helped by the previous Unix culture.  
The Linux universal operability image is here reinforced since the evolutions of its 2.4 kernel supporting 64 bits Intel Itanium processors, SMP (Symmetric Multi Processing) clustering, plug and play features and USB (Universal Serial Bus) connections.  
The commitment of traditional constructors to Linux will reinforce the global public sector confidence, (but it will also strongly reduce business for early Linux hardware integrator as VA Linux)
- The gradual discovery by IT managers that Linux can be an alternative database platform (with the OSS MySQL or PostgreSQL as well as with the proprietary Oracle, DB2, etc.), an ASP or middleware platform (Enhydra, Zope) as it is an Internet of file server platform.
- Experiences from the last two years concerning TCO, reliability acquired with Internet services, and coexistence between proprietary and open source software in LANs and Intranets will be reported by OSS pioneer users and will bring more public sector IT managers to consider OSS. The media coverage and the revelation of successful implementations will create an important, although not immediate, snowball dynamic. This dynamic is especially important because the lack of presence of Open Source Software is mainly not related to technical excellence issues, but to a lack of general, philosophical sentiment of acceptance at managerial level.
- Political support will generate a growing pressure in order to equilibrate environments, to require open standards and to admit open source based solutions in call for tenders. The FUD effect related to the US and EC actions against Microsoft and its possible splitting has also an influence.

- Continuous support from Free Software advocates. This is one of the main reasons why the Open Source environment (Mainly GNU/Linux) survives where other proprietary operating systems efforts (including UNIX) have failed in the past. The open source character itself provides and continuously renew an army of enthusiasts that nobody can buy or match. This continuous commitment makes more for Open Source than the “opportunity alliances” now offered by the IT industry

Linux's close future is obviously in servers and in embedded devices, where acceptable revenues may be produced by integration services, and where most of investment from IT industry giants are now concentrated.

## **Growing need for Application Service Providers (ASP's)**

The ASP provides server-side applications available to anyone with an Internet connection (a thin PC client, with a browser). The most common type of ASP offers data storage, but others allow one to use complete office style products. The OSS ASP solutions encounter a growing popularity with Zope, or Enhydra, and may become an interesting alternative to “fat expensive complex and always more powerful” clients, with an also quite “expensive and complex office suite” running on each of these clients.

The Office IT manager should then carefully considers the ASP possibilities with some restrictions:

- The network high-volume traffic and server availability must be granted: as it was the case with old mainframes and their non-intelligent terminal a too slow network, any network or ASP server downtime will cause a general paralysis.
- The ASP solutions are therefore more adapted to fixed desktop installations than for mobile or autonomous users.

According to these restrictions, ASP solutions can procure major advantages in reducing the size (power, RAM and need for upgrades) and the administration cost of desktop non-mobile workstations, for example in education or inside stable administrations

## **The move to Mainframes**

IBM and other major constructors based a significant part of their business strategy on major Linux push in 2001 (following the declaration of IBM CEO Lou Gerstner to spend \$1 billion on the Linux operating system developments in 2001).

For similar reasons of simplification as the growth of ASPs, the motivation of these investments is that, for traditional constructors always interested in producing high-end hardware, OSS gives a new chance to mainframe (and perhaps make the high cost of their mainframes more acceptable).

Provide broad band powerful networks are available to support the increased traffic, a main frame central Internet/Intranet application sever with “thin remote clients” (basic PCs or terminals with an embedded Internet browser) will cost less in system management than a sophisticated client-server architecture (fat clients with higher license costs and decentralised upgrades and management).

## Desktop domination

The fact: Desktop market is dominated by a leading vendor: Microsoft.

The key of this domination is not so much the Windows environment alone but the fact that it is integrated with MS-Office<sup>12</sup>: MS-Word, Excel, PowerPoint, Outlook, Access, Project etc. are the reference tools for an entire generation of users.

The success obtained on server has given ideas that OSS can also prevail on desktop, but previous failures (Corel, Novell, Lotus with commercial challengers or OSS developer like Eazel<sup>13</sup>) make it clearly uncertain.

By continuously improving and adding new and complex functionalities to its operating system (for example a CD writer, instant messaging or multi-media tools in Windows XP) and to its Office suite (for example, a multi-lingual translation system in its Office XP), Microsoft has good chances to continue to dominate the desktop market. It will even take substantial parts of new markets, as the PDAs with the Pocket PC, ASP and Web servers with .NET, videogames with the Xbox etc.

Facing Microsoft's continuous extension (by improving reliability and user friendliness, but also by introducing new functionalities that may also create new interoperability problems when the resulting documents will be exported on other systems), the Open Source Software developers cannot much more than react, being technically always one step behind. In this domain, the "follower" reputation of the OSS community is not an intrinsic characteristic, but the consequence of the main vendor domination.

This "one step behind" syndrome is translated by the reluctance of the majority of users to be forced to learn a new (OSS) system, with no perceived profits and worse, interoperability problems with office documents.

Despite the quality of GUI desktop interfaces as KDE and Gnome, most of the OSS investments are now dedicated to servers, not to desktops. This is likely to continue to be the case, as the only way to gain a part of the desktop market is a massive growth of developer's community, generating a massive growth of users that will generate more revenue for Linux vendors, trainers and integrators.

So, if no commercial company, not even Open Source free products can really compete with Microsoft on the desktop market, what are the evolution factors?

- The market expectation for open standards and stability;
- The political requirement to construct e-Government on non proprietary basis;

---

<sup>12</sup> In Microsoft's fiscal 2001 third quarter, Office accounted for 37 percent of revenue--more than any other product. That is however a decline from 40 percent in the second quarter and about 46 percent during fiscal 2000. The new licensing policy announced in May 2001 intend to impact that decline.

<sup>13</sup> On May 15, 2001, Eazel – developer of Nautilus, a graphical shell developed as an integral part of the [GNOME](#) desktop environment - has ceased operations, as efforts were not sufficient to secure additional funding

- The anti-trust and law suits
- Microsoft pricing and licensing policy, if finally perceived as too expensive for a certain class of users (SME's, Public Sector, Emerging countries like China, some individuals)

More specifically, the following elements may be considered:

- The growing and “excessive” complexity of the dominant desktop tools, and the frequency of their upgrades (Office 95, 97, 2000, XP) may lead IT manager to react by adopting a more stable environment.
- A growing pressure from the political actors (parliaments, governments, European institution) for more diversity and therefore, a recommendation for alternative desktop tools in wide parts of the European or national administration. The requirement to operate an open office productivity suite on multiple platforms, including Linux, the Solaris operating environment and Windows, may then lead to the adoption of the StarOffice or similar.
- The licensing price and the Microsoft's activation policy.  
In May 2001, Microsoft revised software licensing, raising upgrades between 33 percent and 107 percent, according to Gartner. A significant part of business customers (80% according Gartner) could face the issue of upgrading to Office XP before Oct. 1 or paying a heftier purchase price later on<sup>14</sup>.

Gartner estimates that more than 80 percent of Microsoft customers will have to decide whether to upgrade to Office XP before the deadline.

More generally, as it was recently the case with the Oracle pricing policy, many users may re-examine their strategy at the occasion of Microsoft's changing licensing programs: the license price policy (and license activation) will lead IT managers, especially in SMEs, to consider more seriously free alternatives.

If maintained as announced, the Microsoft' activation policy may finally be a good point for everybody: it will improve the market honesty by limiting frauds, and make possible a better market evaluation. It will also indirectly reinforce the use of Open Source office suites, as unauthorized copying of MS Office XP will become more difficult.

---

<sup>14</sup> The upgrade situation applies to all Microsoft products, not just Office. For example, businesses using Windows versions before 2000 or XP before the deadline would have to upgrade to one or the other to join Software Assurance.

Together with the above factors, the promotion of more adapted (to specific needs and public sector requirements), easy to install, distributions and the evolution of both OSS Office suites and file format standards will be determining here. When – as it is announced for later this year with StarOffice version 6 – OSS will be able to be deployed side-by-side with MS-Windows with more compatible file formats with all versions of Microsoft Office, including file saving in the recently finalized XML file format specifications, and launching its various components separately with less resources than actually (and not all at the same time as in the Star Office 5.2 version), it will present a more persuasive alternative. The arrival of new personal applications (E. g. Gnu-Cash in finance) is also important.

## **The All or Nothing Change on desktop**

On Desktop, the adoption of OSS based solutions cannot be “Hidden” or “transparent for the user” as it is for servers. It hard to think that in a 25.000 workstations integrated department, 50% of the users will run MS-Office and another 50% will use Open source.

Dual boot installation that have been used by IT managers to introduce OSS and to make reluctant users more familiar to them without breaking with the MS-Office world are not a general solution either, since it make the desktop administration and use more complex instead of more simple.

The consequence is that the desktop environment choice will be strategically decided from the top, for entire departments, both for licensing cost and political reasons. Decisions will take time and pain to be taken (training and data migration may be expensive) but changes once decided, will be brutal, as soon a real alternative to MS/Office will be offered.

For these reasons, we do not believe that the desktop challenge is definitely lost for Open Source. The server battle takes most of the energies, and push the desktop side in the background, but it will soon come back.

The declarations of a dead Linux desktop are proving to be premature. Significant moves may occur in reaction to Microsoft's price and licensing policy, but there is still a serious work ahead before GNU/Linux desktops and their associated applications become generally recognised as a credible, user friendly and stable alternative.

## **Service market**

### **Service provider groups**

1. The Linux distributors (distribution providers that include also guarantee and support on request). They started by offering support to end-user individuals, and they are now moving to business packages including annual contracts and high-availability support that should be more convenient for public sector and enterprises. They are of course specialised in their own (mainly GNU/Linux) distribution. In Europe, the two main distributors are SuSE (Germany) and Mandrake (France).
2. The system providers, which include OSS operating systems (mainly Linux) on their own packaged hardware, but do not distribute their own version of the Operating System. They are IBM, Compaq, HP<sup>15</sup>, etc.  
The strength of these companies is their strong installed client base, often based on previous proprietary Unix: They are naturally focused on Public Sector and large/medium enterprises and can use their established good reputation to ensure support, mainly when migrating their hardware from proprietary Unix to Linux.
3. The global integrators, which cover the complexity of existing infrastructures – including proprietary and open source components. They are rather oriented to large clients and maintain usually teams of several hundreds of consultants in several European countries, with a wide panel of competences including open source solutions. They are IBM, Unisys, Accenture, CSC, Cap Gemini E&Y, PWC etc.  
The strength of these companies is their reputation in managing large projects and their capacity of understanding the constraints of existing infrastructures, of analysing the needs and deploying “compound / heterogeneous” solutions. With their stronger financial assets and management capabilities, they can also act as prime contractor for large projects or within frame-contracts and use more specialised little companies as subcontractor.
4. The specialised companies that are strictly OSS service providers. They are usually smaller and more recent companies or start-ups, with dynamic development and less bureaucratic culture, but they are also less experienced in large project management and integration. They present the advantage to be vendor neutral and to support several Gnu/Linux or BSD distributions. They are, for example, Mind (in Belgium), Alcove, Arkane, Atrid, Aurora, Linbox etc. (in France)<sup>16</sup>, Thiesen, Abas software, Sernet, Linux Partner tec (in Germany)<sup>17</sup>

---

<sup>15</sup> IBM has embraced Linux and OSS to the extent it is a selling argument for a new generation of e-business platforms and CEO Lou Gerstner announced in December 2000 that IBM will spend \$ 1 billion on Linux development. On hits s/390 mainframes, it says that “Linux is bringing more flexibility than ever before”. IBM chose the open-source [Apache](#) WebServer to support and bundle with its WebSphere suite. It has since released the Secure Mailer in open source and launched the [AlphaWorks](#) site to disseminate cutting-edge IBM technology in source. HP has started a move to Open Source in December 2000 and hired Bruce Perens (from the Debian / OSI projects) as Linux strategy manager. Inside a new Linux Systems Operation division, HP develops Intel Itanium IA 64 Linux based multiprocessors.

<sup>16</sup> See French enterprise list on the MTIC site: [http://www.mtic.pm.gouv.fr/bouquet-libre/aide\\_technique/support.shtml](http://www.mtic.pm.gouv.fr/bouquet-libre/aide_technique/support.shtml)

<sup>17</sup> See German enterprise list on the Berlios site <http://sourcebiz.berlios.de/enterprises.php3>

## Categories of Support

The Support may also be divided in the following categories:

### 1. Installation support

A distributor provides usually a free installation support during a 60 to 90 days period

Mandrake Soft for example gives a 60 days support by the “Mandrake Expert” service, plus 30 days support by phone.

Mandrake provides also an online support with guided question entry. The site proposes a consulting service, installation and integration services, made on measure developing and a list of partners (ISV - Independent Software Vendors, VAR - Value added resellers, and OEM - Original Equipment Manufacturer)

SuSE provides differentiated installation support for its personal 7.2 version (60 days) and for the professional one (90 days after the date of purchase). Questions related to a previous distribution will be answered no more than 60 days after the release of the newer version.

In addition, SuSE provides the support of its database with keyword searching, FAQ, History of development, thematic and version approaches.

### 2. Support packages

Distribution companies offer packaged support named “Incident packs” or “Call packs”, allowing the processing of a number of “problems” (a certain number of problem processing are sold in one package) SuSE for example, offer a panel of 1, 5, 10, 20 and 50 calls to commercial customers (“Support Angebot für Gesellschafts”).

For private users, prices are lower, but the service is limited to 30 minutes per call, from 9 to 18 hours – working days only – at a price of 46,60 euro (1 call) to 208,80 euro (5 calls pack at 41,7 euro per call).

For enterprises, the pack price starts at 240 euro per question.

Call pack however were never very popular in Europe, and offerings are reduced by a growing demand for annual maintenance contract

### 3. Annual contracts

The types and features provided in annual contracts vary from a vendor to another and are also tailored according to the client specific needs, size of installation, level of requirements. Typically, hardware vendors and OSS main distributors proposed these contracts to business and public sector clients and they range from standard business hours coverage to full 24 x 7 coverage for high requesting applications.

Mandrake and SuSE propose different types of contracts, on 3, 6, 9 or 12 months.

The content of the contract varies in term of:





- **Coverage hours and days:** 10x5 to 24x5 or 24x7, working days only or all 365 days / year.
- **Reaction time** ranging from 1 hour in the best case to 8 or 16 (or “next business day”)
- **List of supported products** Hardware and/or Software
- **Personalization** - Inclusion of individual personalized consultancy and auditing
- **Patch- & Update management** (in case of new version)
- **1<sup>st</sup> / 2<sup>nd</sup> / 3<sup>rd</sup> Level Support** definition and coverage
- Supported infrastructure type, from **Desktop-PC** to **Mainframe**

#### 4. Specific developments and software integration

Most of the large companies offering global project management and software integration services can be asked to propose strategic consultancy, development and product integration. Developments may be project based (usually for a fixed price in Public sector, more rarely on time and material price), or frame-contract based (the frame contract usually specify the man/day prices for each category of profile, as the procedure to fix the number of days each time a new mission is launched by the contracting authority). The service includes also personal training; help desk and global support based on specific SLA (Service level agreements).

#### 5. Open source free support

Last but not least, and although it does not appears from statistics, the most typical and - according many users - the preferred and most efficient form of OSS support always come from the various Open Source developers communities.

The Internet allows programmers and technology specialists from non-related companies to collaborate while working on a problem: because the code source, like a book at a public library, is open to everyone and can be modified and tested by anyone, “you put a question on the Web, and you get 15 answers back”.

“The advantages of open source, such as price and access to source code, are compelling,” Merrill Lynch analyst Steven Milunovich wrote in a research note. “Meanwhile, the objections are falling one by one.”

Although it was first said that it had “no direction” the efficiency of the OSS development and support model is now acknowledged world wide even by competitors: “We recognize that OSS has some benefits, such as the fostering of community, improved feedback and augmented debugging<sup>18</sup>”.

The only question is “will that form of support and community spirit survive or continue to be sufficient in the future, once and if OSS become generalized in Public sector and business? Will it work as well for high-end government and business activities that it works now between pioneers and advocates?”

---

<sup>18</sup> Craig Mundie, Microsoft Senior Vice President “The Commercial Software Model” lecture at The New York University Stern School of Business - May 3, 2001. The efficiency of the OSS development model was already admitted by Microsoft’s collaborator D. Valloppillil in the famous “Halloween” document - see: <http://www.opensource.org/halloween1.html>



Despite the fact all people revolutions are invariably recuperated by political leaders, business and finance managers, lets believe that – at least at complementary level, something from the free Internet spirit will survive.

## Market size and Trends

Until 2001, it is a common evidence to discover that in specific server domains, OSS covers 25% and more of the market in volume, but generate only 1% of the revenues. This is because the market is still in its infancy, and because at the beginning OSS were installed by self-supporting “hobbyists” on specific and improved systems requiring few external and billable support (for example the Apache Web server or the SamBA file server) and few support at user desktop level.

The growth should occur mainly due to a generation of customer using OSS (Linux) as they did for their previous general-purpose Unix, and requiring a higher level of contractual liability and maintenance outsourcing thanks to annual support contracts and high-level help desk for broader non-specialist user panels.

From 2000 to 2004, IDC<sup>19</sup> forecasted a rapid growth (global market in \$), reaching 10 times the 2000 market size in 2004.

Linux support Service revenues 2000-2004 (million \$)					
Year	2000	2001	2002	2003	2004
Revenue	28,3	56,6	105,0	177,0	285,0
Growth (%)		97,9 %	87,5 %	68,6 %	61,0 %

For strategy or business reasons, a number of actors of the IT industry have announced during the last two years their support to the OSS movement.

## OSS Model evolution

---

<sup>19</sup> Source: IDC Linux support Services, Forecast and Analysis 1999-2004 – by Carol Monaco and Ana Volpi – IDC 2000 – p. 21

## From freedom to standard

The Open source model, was imagined and promoted by strong personalities or “gurus” like Richard Stallman, creator of the Free Software Foundation, of the GPL and the GNU project, the promoters of the OSI - Open Source Initiative, Bruce Perens, author of the Open Source Definition, and Eric Raymond, author of “The Cathedral and the Bazaar” explaining the developments principles of the OSS model, Linus Torvalds, creator of the Linux Kernel, Miguel de Ices of the GNOME GUI Desktop Project, Larry Wall, creator of the Perl Language, Guido van Rossum, creator of the Python Language and others as Tim O'Reilly, publisher of reference documentation on Linux and OSS.

As soon a revolution succeed, it is invariably recuperated by political leaders, by business and finance managers. The same occurs and will occur with the open source movement. IBM, HP, Compaq, Dell and the major commercial distributions will lead the movement with organisations as the Linux Standard Base (LSB) in order to impose new a discipline to keep Linux “standard” and avoid forking (the Linux fragmentation).

The “right” to fork and to produce original development from the existing basis will be more theoretical, due to the growing power of marketing.

On the other hand, the LSB specification will simplify the developer’s task, by providing a single environment for Linux development and it is to forecast that the next major release of most Linux distributions will likely comply with the new LSB specifications<sup>20</sup>.

## Development model

Why write, package, and give away software for nothing? There are a variety of reasons. Many do it to gain programming experience. A young brilliant student has few chances to integrate from scratch a leading proprietary software developer team, but may well both deliver a winning end-of-study work and gain a relative celebrity in contributing to a major OSS project. Although he will not win immediate money, the respect of his peers (and professors) will provide him moral gratification and the acquired experience may ensure him important gains when finding his next job.

Others may contribute for altruistic reasons, as a thank you for free software they have received, or to make free some nice piece of code they developed into an administration or into their university, with no chance of any practical possibilities to make it commercial. In addition, by opening up the development to the larger community, a computing problem is shared and solved among peers with similar problems. In the end, it is hoped that a more powerful solution will be developed.

---

<sup>20</sup> Version 1.0 of the LSB specification is expected in July 2001

Linux is the best example of a successful open source project obtained by a cooperative development environment, — it was quickly capable of taking on the characteristics of a real Unix operating system with multi-user and multitasking features. In addition, Linux also claims what no other operating system can do— it has been deemed stable and reliable, it runs on more hardware than any other operating system, is scalable and customizable, and is free...

The new tendency in development in a growing intervention, directly or through sponsorship, of industry giants coming from two areas: hardware and services (or both). Hardware maker are finding in open source a “return to the first ages” where their domination and freedom was not threatened by a licensed software vendor (Microsoft). Themselves will loose a part of their revenues attached to proprietary system licences but as they also deliver integration and services, they will recover it elsewhere. Service providers and integrators also will globally win more or no less with a scalable and customisable system than with a closed one.

## **The projects / the project developers**

At the origin, there is generally one individual or a very small group. After a first development (usually corresponding to specific needs) they start publishing it in specialised Internet forums as an OSS “Project”. If enough developers are interested, a project web site is created, a community is growing (an Internet “Noosphere”) and – sometimes without ever meeting physically each other - assembles, develop and test the software.

It is practically impossible to make an exact calculation of all the OSS developers. The number of published projects (from the simple published planning to the most mature solution) is calculated by Source Forge to 8.184 for the 10 most used license types. Our estimation is about 12.000 living projects.

The number of developers involved in each project varies from some units to several hundreds. In the case of the FreeBSD project for example, thousands of developers worldwide funnel their work to a team of about 240 “committer” developers. Based on an average of 20, and according to the hypothesis that each developer works on two projects, the OSS developer community may be evaluated to at least 120.000 active persons, working without administrative overhead...

## **A multiplication of Alliances to support OSS**

### **An Open Source Development Laboratory (OSDL)**

The Open Source Development Lab <sup>21</sup> mission is Providing Open Source developers with computing resources to build data centre and telecommunication class enhancements into Linux and its Open Source software stack, enabling it to become the leading UNIX Operating System for e-Business development and deployment.

---

<sup>21</sup> [www.osdlab.org](http://www.osdlab.org)

Based in Beaverton, Oregon, OSDL results of an alliance between 19 sponsor companies including Caldera Systems, Computer associates, Covalent, Dell, Fujitsu, Hitachi, HP, IBM, Intel, Linuxcare, LynuxWorks, Miracle, Mitsubishi, Nec, Red Hat, SGI, SuSE, TurboLinux and VA Linux.

### **The WBEMsource Initiative**

Created in June 2001, the WBEMsource initiative illustrates a heterogeneous collaboration between companies, consortia, and open source projects to contribute to the progress of interoperability. The goal is to improve the manageability of large-scale Web-Based Enterprise Management (WBEM) implementation. The new group will work to define and promote an open source enterprise management environment based on the Distributed Management's Task Force's (DMTF) WBEM standards.

Organizations supporting the WBEMsource initiative include BMC Software, Caldera International, Cisco Systems, Compaq, Computer Associates, the Distributed Management Task Force, Evidian, Hewlett-Packard, IBM, The Open Group, the Storage Networking Industry Association, and Sun Microsystems.

### **The Linux Standards Base**

The Linux Standards Base (LSB) is a coalition of developers and independent software vendors (ISVs) that is trying to consolidate a standard Linux implementation to prevent the Linux market from splintering in a similar way to Unix. In early 1999, this coalition began working to outline standard operating system facilities (unrelated to the kernel), such as file-system hierarchies and libraries. This group advocates creating a base set of libraries, APIs, and interoperability measures. The LSB includes the major vendors supporting Linux, including Caldera, Corel, the Debian Project, Delix Computer GmbH, Enhanced Software Technologies, IBM, LinuxCare, Linux for PowerPC, MandrakeSoft, Metro Link, Turbolinux, Red Hat Software, Software in the Public Interest, SuSE GmbH, VA Linux, WGS, and SGI.

### **The Open Standards**

Open source software development is not to dissociate from the sharing of open standard, allowing applications to communicate and ensuring solution interoperability.

The Internet operability is based on such standards as TCP/IP, HTTP (The common Internet protocols) and HTML (Hyper Text Mark-up Language).

A prominent role in this field is played by the W3C - [World Wide Web Consortium](#) that was created in October 1994 to lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability. The W3C has more than 500 [Member organizations](#) from around the world and has earned international recognition for its contributions to the growth of the Web.

The common sharing of standards like XML (the eXtensible Mark-up Language), or SOAP (Simple Object Access Protocol) will play a determining role in simplifying the exchange of business data between administrations. Further initiatives as UDDI (specification for distributed Web-based information registries of web-services and publicly accessible implementations that enables the content interchange between different services in the Internet) are expected as the next step to create a uniform service description and data interchange format.

The set-up and maintenance of open standards as the basis infrastructure needed for free development; the limitation of software patents on these standards and the fight against the permanent temptation of monopolistic leaders to recreate proprietary standards that can restrain innovation and interoperability is one of the major preoccupations of the Open Source Community.

## The OSS organisations

Developers are grouped in organisations that are helping developers to communicate and to find resources. Here are some of them:

Berkeley	Berkeley Soft Foundation	<a href="http://www.bsd.org/">http://www.bsd.org/</a>
FSF	Free Software Foundation	<a href="http://www.gnu.org">http://www.gnu.org</a>
Mozilla		<a href="http://www.mozilla.org">http://www.mozilla.org</a>
OS	Open Source Organisation	<a href="http://www.opensource.org">http://www.opensource.org</a>
SF	Source Forge	<a href="http://www.sourceforge.net">http://www.sourceforge.net</a>

In addition to the international organisations, almost each European country have national organisation, organising expos, symposiums, seminars and contributing to the open source culture in public sector and enterprise.

For example, in France : [AFUL](#) (Association Française des Utilisateurs du Libre) and [April](#) (Association pour la Promotion et la Recherche en Informatique Libre)

## Technical co-existence Proprietary/OSS

On servers, the insertion of OSS (for example GNU/Linux based) systems in an existing network may be perfectly transparent for users. A SamBA file server is compatible with the SMB Microsoft protocol and the windows users will not see any difference between a NT and a Linux server.

Many examples of this interoperability are related to Oracle (proprietary database) installations, running indistinctly on Linux, NT or other platforms.

On desktops, a co-existence causes more problems: the consistency of the workstation administration and the necessity to preserve a good interoperability between the various users of an organisation naturally leads to “all or nothing” solutions.

Specialists are able to install dual boot workstation with separate Windows and Linux partitions, but apart from developers and the use of specific applications coming out of the two worlds, this dual installation will not correspond to the average office user needs. It requires complex tools and requires from 500 MB to 1 GB extra disk space.

Note:



To run Linux and Windows together from the same hard drive or to run Windows on top of Linux, the two environments will be located in separate hard drive partitions (most distributions need approximately 500 MB of disk space). Whereas Windows uses just one hard drive partition, Linux generally requires at least two. Linux distributions— such as Caldera's— will shrink the original Windows partitions and then create the new partitions for Linux. Others, however, require a utility to be run, like PowerQuest's PartitionMagic first, to get the space ready for Linux. Then, when the machine boots, the LILO utility (the Linux Loader) will play as boot manager, allowing a choice of which OS to run.

An alternative is VMware 2.0 allows Windows 9x, NT, or Windows 2000 to run on top of Linux— or at the reverse Linux on top of Windows NT or Windows 2000. This is obtained by opening a "virtual PC" inside the selected main operating system. VMware allows the user to install a secondary OS on the virtual machine. Another package, GraphOn Bridges, lets users run Windows applications over a network.

## OSS and Innovation

Is OSS only based on the developer's dreams, when discovering a proprietary tool, that "It would be nice if it was free" and – afterwards, on the resolution "Let's make it free"? Is the OSS development model "reactive" rather to "innovative, and will it kill innovation?

It is a fact that OSS was originally a reaction against proprietary software (With the Richard Stallman's Free Software Foundation and the GNU project reacting against proprietary Unix). From Linus Torvalds writing his own kernel up to the last developments, OSS is often developed to provide free alternatives to already invented solutions.

Similarly, the open source community did not invent the Internet (although key chunks of the Internet itself are open source), but they invented its collaborative method and multiplied its efficiency: their genius is in creating a new, original development model and in making free and for everyone solutions that were not originally conceived for that.

The open source project panel analysis partially explain the controversial remark of BEA Systems founder and Chairman and Chief Executive William Coleman<sup>22</sup> that *"open source is the end of innovation ... because it can't happen until it's so broadly understood what's going on that the innovation has slowed down to incrementalism."*

The reality is more complex. In Europe, research institutes and laboratories (as the INRIA in France) have based a good part of their innovative developments on OSS.

---

<sup>22</sup> also former vice president of Sun Microsystems

Based on the free aggregation of existing components (without being limited by restrictive licenses or patents), the OSS development model may on the contrary really accelerate innovative developments.

A majority of the global 2.500 US companies have the same opinion, that the two biggest effects Open Source Software will have on the industry are “Force competition” (52%) and “More innovation” (50%) instead for example of “Lower prices” (14%). “Open source helps innovation because it provides the opportunity for more people to participate. As a result, more ideas develop,” said an aerospace company<sup>23</sup>.

But, even when reacting to commercial success, will OSS restraint innovators to start really original developments?

The natural and human tendency is that innovators wanting to become rich will continue, in most of the cases, to try to release their innovations under proprietary or commercial licenses.

At this stage, and apart from exceptions, two main hypothesis lead to the development of new open source solutions:

1. The commercial success of the innovation leads the open source community to react and to develop a free alternative flavour. Until now there is no evidence that any really commercially successful solution has ever been killed by its OSS alternative rather than by other commercial competitors;
2. The commercial failure of the innovation leads the innovators to give a second life to their solution, by providing the source code to the open source community and changing their business model from licenses to services. The open source model provides then a chance that the concerned innovation will not be lost forever and that new developments, perhaps still more innovative, can start from it.

Another consideration concerning innovation is that the viral character of the GPL (the FSF General Public Licence that governs about 80% of all OSS projects) will – at the end – contaminate all existing software<sup>24</sup>

But this consideration also looks excessive, as the open source community do not register patents that should really restraint innovations from commercial companies. In term of Intellectual Property, the OSS GPL community requires the respect of its copyright: if you use a piece of code “As is”, then the resulting product must be GPL also, otherwise – and you will do it if you are really innovative – write your own code. The fact that the GPL is definitely chosen as distribution license by a great majority of open source project developer (more than 80%) and the fact that this enormous factory of project is exponentially growing demonstrate on the contrary that OSS in general and the GPL in particular permits a greater rate of innovation, with greater efficiency in terms of scarce resources.

---

<sup>23</sup> Forrester report, p. 4

<sup>24</sup> this idea was launched by Craig Mundie in May 2001



The conclusion is that the open source development model was indeed rather “reactive” towards commercial dominations, but that it will not kill innovation. On the contrary, it can stimulate innovation. It will not kill proprietary software either concerning really original solutions. It may rather limit the profit margins of well-established vendors, making the improvements, new releases and functionalities they bring to their commercial solutions more questionable.

## The Business model

Is there a specific economy of OSS?

In recent years, open source has become accepted in commercial environments, where profit has begun to drive the market. Thousands of commercial entities make money on open source. Organizations make revenue from services, support, and proprietary add-ons and extensions. Vendors, including Red Hat, IBM, and Apple, develop Open Source products as loss leaders for services, support, and proprietary add-ons. Several of the more common business models for open source software include these<sup>25</sup>:

- Distribute open source products on packaged CD-ROM, such as with Linux distributions Red Hat, Caldera, and others.
- Add value to open source products— this could be added software, hardware, training, support, or services to an open source product.
- Provide open source software to promote related products— some vendors seed the market with software, which expands the market and drives it toward other products and services offered by the vendor. Examples include Netscape Communicator and Microsoft Internet Explorer. Netscape is hoping its Communicator product will encourage sales of its server software and portal service.
- Homegrown software development— not for commercial profit. Using a team of developers over the Web (the same framework as Linux's development), different organizations that share the same problems can cooperatively build software that can work for all.

---

<sup>25</sup> On business model and perspectives, see: Gartner Group (Mary Hubley) – Technological survey – 11 April 2000 “Open Source Software: Perspective” (DPRO 90332)- and 2 January 2001 “Linux technology perspective” (DPRO- 90279)





According to Proprietary business model advocates, when comparing the commercial software model to the open-source software model, one should look carefully at the business plans and licensing structures that form their foundations. This comparison leads to the conclusion that the commercial software model alone has the capacity for sustaining real economic growth. Intellectual capital has always been, and will remain, the core asset of the software industry, and of almost every other industry. Preserving that capital--and investing in its constant renewal--benefits everyone.

The most successful OSS vendor's response is not in contradiction with proprietary competitors: indeed, a commercial model only has the capacity to ensure economic health, but, "*we believe we can deliver a superior value proposition with an open source model, and that people will pay for that superior proposition in preference to an inferior one. That we choose to protect our product with the GPL speaks to our evolution as a modern software company.*"<sup>26</sup>

There is no specific OSS economy, in the sense that economic rules governing OSS are the same as economic rules governing all economy: at the end, people must get paid for the work they deliver, and nothing is never for free. There is a specific economical model, in the sense that it corresponds to the OSS TCO/TBO model: the costs and their corresponding revenues are moving from **licensing to services and human development**.

---

<sup>26</sup> Michael Tiemann, CTO of Red Hat, provider of open source solutions

# The OSS TCO analysis

## TCO notion

Total cost of ownership (TCO) is a key measure to determine the real value of a project.

Be TCO measurement, we must not only evaluate and compare the direct expenses, but also the labour and all kind of indirect costs:

Direct costs, composed by:

Assets (best known)

Hardware: - servers, clients, peripherals, network

Software: - operating systems, applications, utilities, management

Upgrades, Supplies, Spares

Labour (less well known)

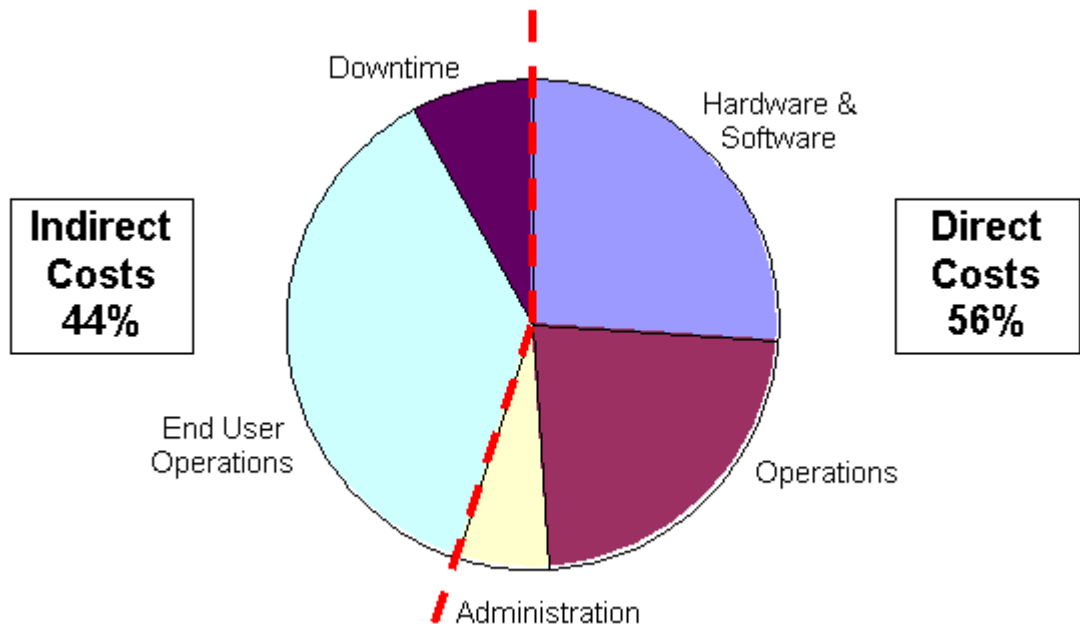
Operations: - technical services, planning and process management, database management, service desk

Administration: - finance administration, IS training, end-user training

Indirect costs and labour (least well known)

End-User Operations: - peer support, casual and self-learning, formal learning, application development, file and data management, futz factor

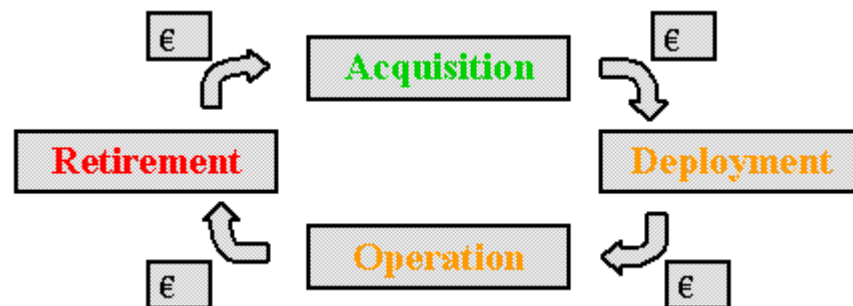
The global cost panel (direct + indirect) is estimated as follows:<sup>27</sup>



<sup>27</sup> Unisys TCO calculation in a public education project - Australia

The above costs should be considered during the whole life cycle of each project, and not only during the purchase or even the deployment of the solution. The impact of the choice and its correspondence to best practice continue to produce measurable effects during the operational phase (x years) and also during the retirement phase. This is specially the case for data archive, data migration and takeover. If the originally selected system produced many years of archived data, according to proprietary document or file formats, the cost of a retirement (the preparation for a migration and takeover) may be amazingly high.

**Costs are present at each step of the life cycle**



## GVC Government value creation

The cost alone do not provide information on the value of a project: it must be related to the benefit, to the savings and to the Return on Investment (ROI) in order to provide the EVC (Economical Value Creation), or – speaking about government – the GVC (Government Value Creation).

In addition to the cost factor (deliver the required service at the best financial conditions and lower public money spending), GVC is measured by:

- Global internal service quality and consistency  
The internal productivity and performance, resulting by the level of functionalities, security, ease of use, internal interoperability
- Business or government value delivered to the external world (Interchange of data with other government agencies and EU administration, enterprises, citizens).
- Consistency with government ethic and policy (independence, transparency, security, privacy, non-exclusion, non-discrimination, neutrality regarding citizens and enterprise technical choices).

All these factors are summarized in the grid hereafter:

Public Sector TCO evaluation grid

	<i>Item</i>	<i>Issue / question</i>	
<b>Direct costs</b> <b>Study</b>	Feasibility study	Consider the cost of evaluating and testing multiple distributions of OSS	
	Need analysis, functional analysis	Internal or external consultancy	
	Technical analysis	Internal or external consultancy Distribution selection Solution identifications, Benchmark Benchmarking with similar (foreign) services (Identify best practices) Technical architecture and design	
	Financial and political analysis	TCO evaluation; Savings evaluation; Cost doing nothing evaluation; Scaling effect evaluation; Government imperative consideration;	
	Hardware	Purchase new hardware Upgrade existing hardware Reuse old hardware	
	Operating system Packaged software licence fees	Evaluate scaling effect	
	Applications Packaged software licence fees	Evaluate scaling effect. How are Internet users considered (named users / connected users)	
	Personnel cost for Hardware and software deployment	Cost of preparing installation standards (E.g. CD-ROMs with a dedicated, simplified distribution);	
	Custom application development (study, development, test, implementation)	Internal or external resources,	
	Existing data takeover and data migration (from previous systems)	Internal or external resources,	
	<b>Operations</b>	Personnel cost for software deployment and daily management	Cost for preparing and implementing a centralised user configuration management tool
		System operation (personnel for normal system operating, monitoring, backup etc)	
Support Maintenance		1 <sup>st</sup> and 2 <sup>nd</sup> level support policy; Agreement with an external service provider	
Energy and space			
<b>Indirect, hidden costs</b> <b>System</b>	System downtime (lost productivity)		
	Technology locking and team	Is the solution motivating the team?	



<b>Operations</b>	motivation	Producing creative solutions?	
	Cost moving to other technologies and platforms	(Loss of know how and culture, departures if technicians and developers dislike the new technology)	
	Indirect Upgrades	Third party tools that may be upgraded too, as user management tools, backup and disaster recovery, storage management, security management	
	<b>End user operations</b>	Training required	Formal training with teacher, other planned education
		End user performances, response to the needs, functionalities	Is the end user application fast and intuitive? Does it fully respond to the needs? Is the time to open, close, operate and save data and programs optimised?
		Peer support, casual and self-learning. Futz factor	Are all functions easy and self-explaining? How much time is spent futzing around, looking or asking for solution?
		Costs for having non standard configurations	Daily cost related to interoperability problems, conversions, adaptations
<b>Government factors</b>	Independence,	Is the government keeping its freedom to migrate solutions and data to other platforms and software, at lowest possible costs	
	Transparency, security	Is the source code available, Can the developer's team check for black boxes, backdoors etc.	
	Long term conservation according to legal obligations	Is the data archiving format secure for long term storage? Will it be readable/usable in 10, 20 years?	
	Interoperability through respect to open standards	Is the document and file format compatible with standards allowing data exchanges with other administrations / actors	
	Authentication Privacy, encryption	Is the solution neutral and compatible to non proprietary standard in these domains	
	Neutrality	Is the solution respecting the neutrality and plurality of choice when used by other administrations, enterprises, citizens	

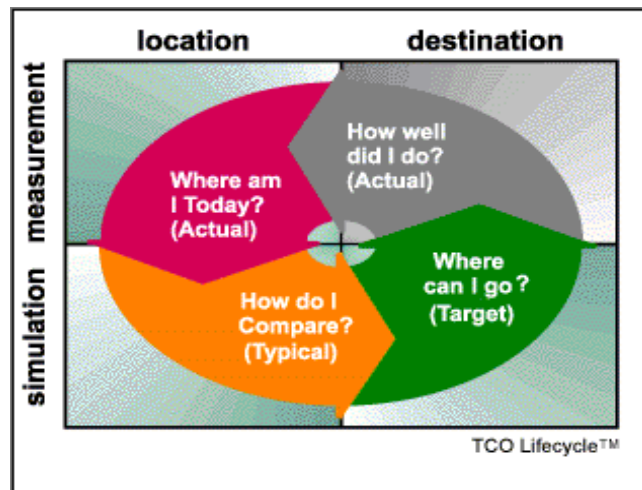
In addition to the costs and government imperatives, the possible savings resulting of the selected strategy must be evaluated as the impact of the scaling effect if the original project is extended to more servers or to more users.

Savings related to	Hardware	Avoiding the purchase of higher cost
--------------------	----------	--------------------------------------



the solution		hardware Giving a second life to existing hardware
	Software	Avoiding costly migration, if planned / expected / forced
Savings / costs related to future scaling effect	Hardware /Software / Team	What if the current pilot solution is extended to 200, 500, 2000 users? Is the development / software distribution re-usable?
Cost doing nothing	Hardware /Software / Team	Taking in consideration the need evolution, the volume evolution
Impact on other IT strategies	Hardware /Software / Team	E.g. the savings/costs of outsourcing less strategic, older applications or services (e.g. Legacy applications running on older systems)

The TCO/GVO estimation itself must be reworked continuously during the lifecycle of the different solutions:



As soon as a “destination” is reached, it must be evaluated (Where am I today?) and benchmarked with the permanently evolving best practice (How well did I do?). This diagnostic is followed by the two questions:  
- Where can I go next?  
- How can I achieve the new deal?

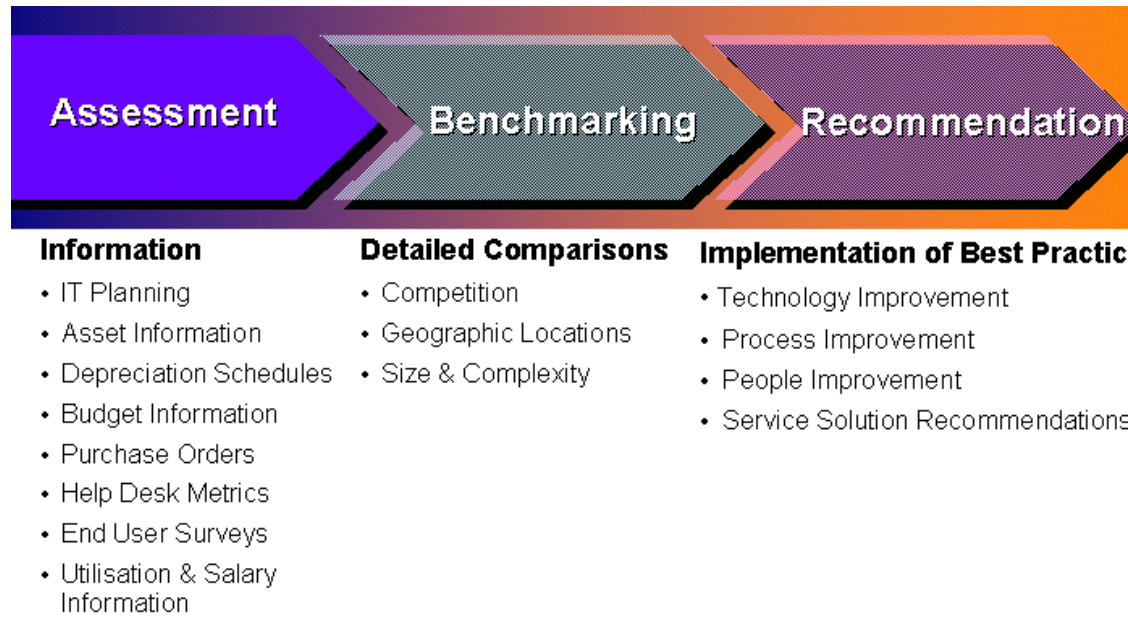
## Benchmarking with Best Practice

The simple fact to select a strategy (open source or proprietary for example) is one of the TCO / GVC components, but the same strategy can lead to very different results depending on qualitative, non measurable factors as the competence and the motivation of the team, the design of the project, the response to the needs, the good preparation and introduction of the solution in a complex, real working environment. Concerning all these factors, the comparison with existing best practice is the optimal evaluation tool. Before to decide for a strategy (e. g. regarding Open Source) public sector IT managers should respect a three step process “Assessment / Benchmarking / Recommendation”:

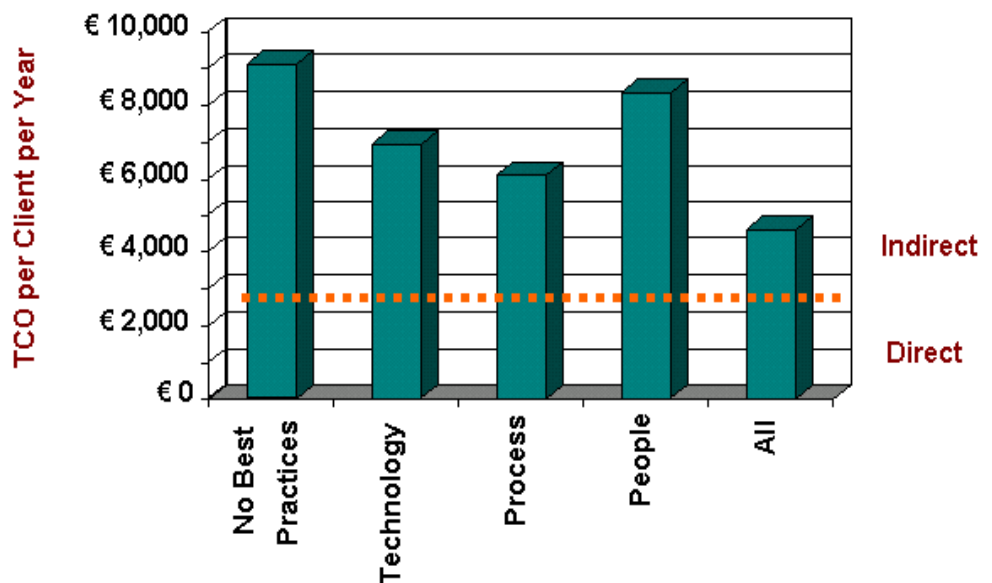
In the picture below concerning enterprise strategies<sup>28</sup>, benchmarking is measured against competition, in similar geographic location, and in enterprise of similar size and complexity

Public sector will benchmark with:

- Similar services in other Member States (rather than competition in similar geographic location)
- Similar complexity and size (number of servers / workstations)



According practical experiences, the fact not implementing solutions according the best practices is the most cost-producing factor concerning indirect costs



<sup>28</sup> Unisys TCO evaluation methodology

## Is general TCO estimation possible “in the labs”?

A TCO / GVO estimation is only possible by knowing and evaluating all components of the specific situation, and all the services a government agency is supposed to deliver, for its own administration, for other agencies, for enterprises and citizens. In most of other cases, laboratory benchmarks or simulations can just provide indications. Simple cases exist, as for example Web server comparison, but some studies are deliberately oriented in favour of one of the tested solutions, and even in clear performance comparison between to similar server configuration, running to different OS, there can be considerable controversy about speed, performance, and of course about TCO.

Laboratory benchmarks generally give few indications concerning TCO, as they are focused on pure performance. They must be considered with prudence, taking in account all conditions (including who commissioned the study) and the reputation of the tester.

Example of these performance contests: the Mindcraft study<sup>29</sup> (1999).

The Mindcraft (April 1999) study awarded Windows NT4 running Microsoft Internet Information Server against Linux running the Apache web server. The study was commissioned by Microsoft, and hosted by Microsoft at the Microsoft Campus. It claimed the Microsoft solution performs 2.3 times faster than the Linux solution. After the study was published, it produced an important controversial literature and Linux experts said that an in-depth analysis of the study showed the Linux/Apache combination was configured to run slowly, and the Microsoft solution was highly tuned.

Facing this controversy, Mindcraft made another “Open benchmark” in June 99, producing new results still giving a clear advantage to NT (1,4 to 2,7 faster), but also admitted “*Mindcraft made some Linux configuration and tuning mistakes in this benchmark. However, the original results and the Open Benchmark results are close.*”

Benchmarks as the “Mindcraft” however, are useful for the developers and users communities as they generate a strong commitment to understand and solve possible weaknesses and bottlenecks.

A TCO oriented benchmark (The BRG study, published by Microsoft in March 1999<sup>30</sup>) was done between NT and a proprietary Unix (Solaris). Not surprisingly, the Microsoft OS was here the winner.

---

<sup>29</sup> Published 13 April 1999, see: <http://www.mindcraft.com/whitepapers/nts4rhlinux.html>

<sup>30</sup> See: <http://www.microsoft.com/ntserver/nts/exec/reviews/BRGTCOWP.asp>



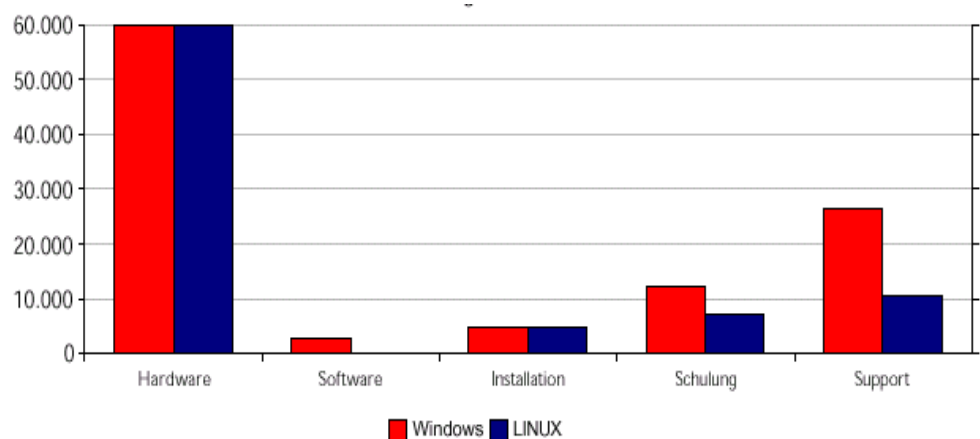
<b>BRG study: 925/990 users (costs in \$)</b>	<b>NT Compaq</b>	<b>Sun Solaris</b>
<b>Initial system purchase</b>	<b>367.250</b>	<b>791.190</b>
<b>Installation, Training</b>	<b>13.300</b>	<b>17.860</b>
<b>System / network upgrades</b>	<b>10.310</b>	<b>17.190</b>
<b>Management costs</b>	<b>439.520</b>	<b>605.990</b>
<b>Application developments</b>	<b>392.870</b>	<b>664.060</b>
<b>Technical support</b>	<b>25.040</b>	<b>40.320</b>
<b>Yearly TCO</b>	<b>1.248.290</b>	<b>2.136.620</b>
<b>Yearly TCO per server</b>	<b>42.130</b>	<b>56.270</b>
<b>Yearly TCO per user</b>	<b>1.350</b>	<b>2.160</b>

Concerning an OSS configuration (Gnu/ Linux) and Windows NT, the German ID-Pro AG<sup>31</sup> provided a “Linux Vendor analysis” in the case of migrating 100 Workstations, from NT to Windows 2000 or to GNU/Linux

This end 2000 comparison attributed lower TCO cost for Linux for both server and desktop.

For servers, the difference came – surprisingly – from training and support (two domains where OSS installations are rather known as “more expensive”). In our opinion, that benchmark may be verified only in environment with an established Unix culture and with adequate management tools.

On Server:



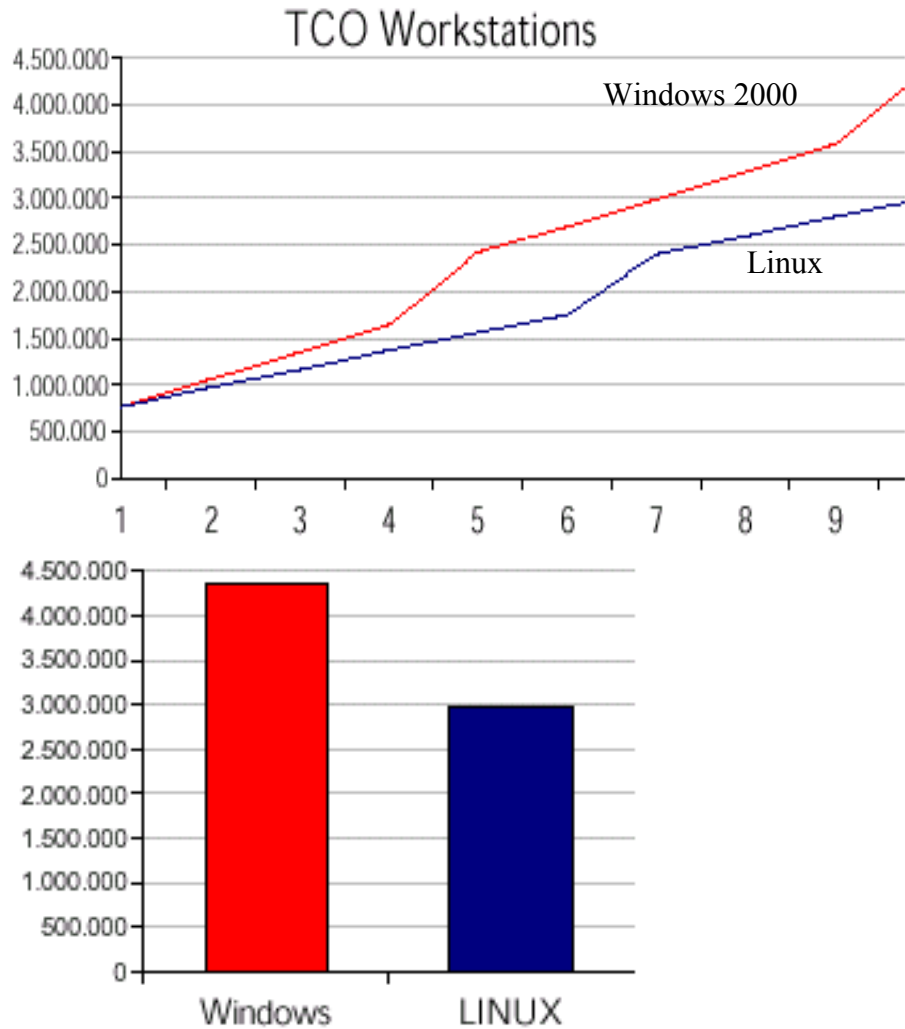
On workstations, ID-Pro AG provided an estimation of about \$ 1.450 yearly per user. It is also one of the sole comparisons between GNU/Linux and Windows 2000.

<sup>31</sup> The ID-Pro AG is a company specialised in Linux Solution, Support, Training – the complete analysis is published at: Link: [http://linux.kbst.bund.de/ws-oss/dokumentation/vortraege/Linnow\\_Kostenvergleich.pdf](http://linux.kbst.bund.de/ws-oss/dokumentation/vortraege/Linnow_Kostenvergleich.pdf)



ID-Pro considers that Hardware, Installation, training costs is identical and put the most severe charges on Windows 2000 in three domains:

- Software cost
- Support (but the purchase of external support should minimise this factor)
- Downtime



Another OSS/proprietary estimation (already mentioned in part 2 of the study) is given by the German Ministry of industry (BMWi), but these figures reflect mainly the purchase and installation costs, which are just one part of the TCO components:

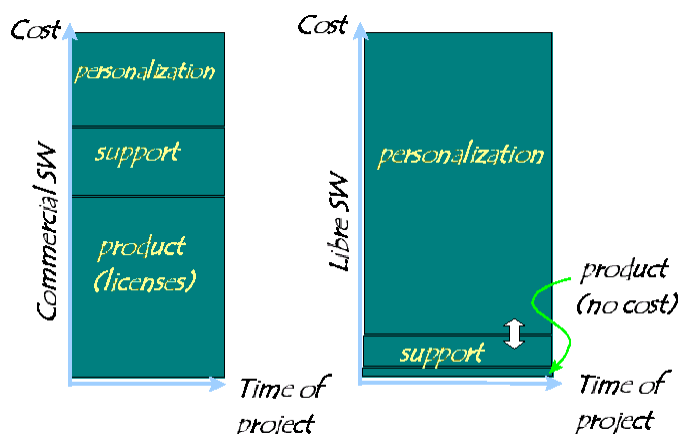
Web Server comparative purchase and installation costs:

Component	Open source		Mixed solution		Proprietary	
	Product / Qty	Price (euro)	Product / Qty	Price (euro)	Product / Qty	Price (euro)
Operating system	Linux distribution	50	Proprietary OS	1.250	Proprietary OS	1.250
Web server	Apache	0	Apache	0	Included in OS	0
Page system	PHP	0	PHP	0	Included in OS	0
Database	Postgress	0	MySQL	0	Proprietary DB	2.145
Installation	8 hours	800	8 hours	800	8 hours	800
<b>Total</b>		<b>850</b>		<b>2.050</b>		<b>4.195</b>

Internet router comparative purchase and installation costs:

Component	Open source		Mixed solution		Proprietary	
	Product / Qty	Price (euro)	Product / Qty	Price (euro)	Product / Qty	Price (euro)
Operating system	Linux distribution	50	Linux distribution (Red Hat)	50	Proprietary OS 15 clients license	1.250 718
ISDN	ISDN – Packets	0	ISDN – Packets	0	ISDN Packets	0
Firewall	Ipchains	0	Proprietary, 25 IP max, 1 year update	2.700	Proprietary	2.700
Proxys	Squid	0	Squid	0	Proprietary	1.217
Messaging	Postfix	0	Postfix	0	Proprietary	2.708
Mail client	Varia	0	Varia	0	Netscape	0
Installation	20 hours	2.000	20 hours	2.000	20 hours	2.000
<b>Total</b>		<b>2.050</b>		<b>4.750</b>		<b>10.593</b>

Finally, a realistic preliminary approach, based on the Spanish Public sector MAP concrete experience is to consider that the global budget is roughly the same, but is used differently, spending more money in services and for investing in people (training) and methodology (creation / personalisation of a dedicated and easy to install OSS distribution) and less in licensing<sup>32</sup>.



<sup>32</sup> Juan Jesus Muñoz Estaban, Case study of the use of libre software in Spanish administration – IDA Symposium on the use of Open Source Software (Brussels, 22-2-2001)

Starting from this assumption that free software (OSS) does not mean automatically low TCO, the response to the user needs and to the government imperatives must be carefully evaluated in each case, according the TCO evaluation grid.

In addition to TCO, Public sector should consider Government value creation in a “Total Benefit of Ownership” approach

A risk-free way to assess the benefits of OSS to particular government agencies would be to initiate a TBO model (E. g. by the Government Control Offices or General Accounting Office. The conclusions of such a model could serve as a road map for future software procurement. The model could address the following questions:

- For a similar investment, does open source software deliver more reliability, security, independence and functionalities relative to its cost than proprietary software?
- Which government agencies could benefit from a transition to open source software, such as the Linux operating system?
- Would it be feasible/acceptable/beneficiary internally and for external “clients” for these agencies to begin a transition to the use of open source software?

## Who should estimate?

Gene Leganza<sup>33</sup> (Giga- Ideabyte) asked: “Who should be responsible for the decision to deploy OSS?”

The answer was that it should depend on the criticality of the technology.

Until now, the approach has rather been Bottom-up: rarely by top managers (CIO’s) but mostly by medium level IT managers, solutions architects, This because the embedding of Linux in a plug and play device, or even the implementation of a front – end IP Apache internet server, has still little impact on the global IT strategy.

The adoption of open source on general-purpose internal servers and even more the extension to desktops is, however a major architecture decision point and must be discussed within the context of the overall technology strategy.

All the above factors must be evaluated to determine the impact on the business goals, the best practices, costs, benefits, risks and flexibility issues.

Only a comprehensive and case adapted TCO analysis can present the priorities and their global value to the Public sector to managers and their technology steering committees.

---

<sup>33</sup> IdeaByte study “Switching to an Open Source OS: Who Decides?” August 11, 2000 by Gene Leganza

# Government action

## Direct support to OSS?

Some of the most spectacular government (US and EU) actions related to open source have been perceived by some analysts as an indirect “negative support”, given by the anti-trust suits (US and EU) against Microsoft.

As Nathan Newman mentioned it<sup>34</sup>, many critics of the Microsoft suit raise reasonable concerns that a purely negative, restrictive approach to punishing the most successful software industry leader might inhibit innovation at the company without necessarily creating a viable competitor. At the opposite, promoting open source software could be a positive policy option that the government might employ to encourage the sort of innovation and competition that is needed to obtain an equilibrated market, without excessive dominance.

It is now a long tradition that the public sector pays for global society infrastructure and commodities as open roads, public education or the police. A public funding and development of the open information society roads – the Internet infrastructure - is now as well admitted as spending billions to construct highways, airports or other public-use infrastructures.

Many argue that because Open Source Software has such public beneficial and may be used as public platform for e-government services, government should promote its use through funding or regulation<sup>35</sup>.

Opponents argue that software is a private industry: the market only should decide and it would not be fair to favour OSS. Some Open Source advocates, as Eric Raymond, have the same opinion for other reasons: they are basically hostile to all public interventionism, bureaucracy and regulation from politicians. They believe the OSS Bazaar<sup>36</sup> will succeed without any help, due to its own quality. Their best advice is “laissez faire, laissez passer”. A public policy (normalization organisations) should be limited to promote compatible standards that serve public needs, and not specific groups or corporate interests, including OSS.

Possible Public sector initiatives are:

- Direct support of projects by funding open source developers (this way is followed e.g. within the IST 5<sup>th</sup> framework programme and by the French government, reserving a part of the FNRS funding for open source projects).

---

<sup>34</sup> The origins and future of Open source Software, - <http://www.netaction.org>

<sup>35</sup> See on this theme, the roundtable discussion issued by The American Prospect – TAP – between Eric Raymond, Nathan Newman, Jeff A Taylor and Jonathan Band ([http://www.prospect.org/controversy/open\\_source/](http://www.prospect.org/controversy/open_source/))

<sup>36</sup> Reference to the famous E. Raymond book “The Cathedral and the Bazaar” describing how the anarchic cooperative OSS development model (the Bazaar) had proven its capability to match the organized, planned and hierarchic software development strategies of IT giants (the Cathedrals).

- Direct promotion of open standards. As open source and open standards are living together, any specification of the use of determines open standards (provide they are clearly defined and identified) will give more chance to open source. The government should more vigorously lend its support to the open standards developed by industry, such as the Internet Engineering Task Force's standard set. Open source and open standards go hand in hand. Open communications protocols and standards of compatibility facilitate OSS development, as they form a fundamental building block of any OSS project.
- Indirect support to the developers: any funding for basic research in scientific areas will indirectly facilitate the production of open source (as a good part of OSS value is produced indirectly, by people working in universities). However, the exponential growing number of projects does not grant their use into the public sector: for that, service support companies should also (although pure OSS developers dislike support and marketing organizations) be awarded with contracts.
- Ensuring a correct, adapted, non excessive legal framework is another indirect protection: one of the main requirements of developers is not to set up bureaucratic organisations, to submit paper proposals, to administrate funds, produce reports and so on. It is to program freely, without permanent scare from lawyers complaining for possible patent law violations or impeaching them to undertake reverse engineering processes when they just want to ensure interoperability with proprietary environments.

So, the government action concerning the legal framework may be:

- o Limit negative effects of intellectual property laws (mainly Patent application), especially when it concerns interoperability
- o Reinforce the positive effects of antitrust law in case of excessive dominancy
- o Avoid new regulations that may directly or indirectly damage the developers freedom

Another possible government action involves the vast pool of software created for internal tasks within governments. Collecting non-classified source code in a series of repositories for the purpose of allowing public access would benefit both government and the public. Additionally, if some individual or group takes an interest in improving some piece of software in use in a government agency, the agency will reap the benefit, at no cost to taxpayers. Long-term security will be improved, by transparency.

## Support to interoperability



## Standards and best practices:

Various “non-governmental” standardization groups care about interoperability. In the IT industry, where the adoption of official norms is generally too slow, these dynamic organisations (as the W3C, the IETF) have taken a dominant role, especially concerning the Internet.

The organizations have complementary objectives. The leading organisation is the W3C. Others are more specialized but collaborate (e.g. Oasis is member of the W3C, which collaborates with the official ISO - International Standardisation Organisation)

Organisation		Founda tion	Members	Role
OASIS	Organization for the Advancement of Structured Information Standards <a href="http://www.oasis-open.org">www.oasis-open.org</a>	1993	171	Interoperability of applications – Industry standards
OMG	Object Management Group (includes MDC- Meta Data Coalition) <a href="http://www.omg.org">www.omg.org</a>	1989	800	Interoperability of middleware end interchange between enterprises
OAG	Open Application Group	1995		XML transactions
Isoc	Internet Society	1991	150 + 6000 individuals	Internet
IETF	Internet Engineering Task Force <a href="http://www.ietf.org">www.ietf.org</a>	1986	150 + 6000 individuals	Internet
W3C	World Wide Web Consortium <a href="http://www.w3.org">www.w3.org</a>	1994	510	Web Architecture

In Europe, the three European Standards Organizations (ESOs) are the CEN (in particular the CEN/ISSS for Information Society Standardization), the CENELEC and the ETSI, that have the major asset of being entirely open and thus free from the pressures of competing commercial interests. Their core values are the creation of products of economic value, based on voluntary consensus of the participants and taking full account of the views of all interested parties.

The characteristic of European norms is that they are systematically transferred, without modifications, in national norms of all Member States: previous national norms are replaced

The disadvantage of the European normalization is its slow decision process that is not really adapted to the fast moving IT industry. About application interoperability and response to the e-Europe objectives, the ESO “Rolling Action Plan” (28 November 2000 – version 3.2.2) provides limited answers, mainly concerning a faster Internet, security and smart cards. Concerning the OSS, it is just written that: “*Furthermore, certain new areas such as “open source software security platforms” may also be of interest.*”<sup>37</sup>

<sup>37</sup>[http://www.cenorm.be/iss/Major\\_Activities/eeurope/JPG39N507r2\\_eEurope%20initiative\\_agreed.doc](http://www.cenorm.be/iss/Major_Activities/eeurope/JPG39N507r2_eEurope%20initiative_agreed.doc)



European Standard Organisation		World corresponding organisation	Role
CEN CEN/ISS	CEN/ISSS, Information Society Standardization System <a href="http://www.cenorm.be/iss">http://www.cenorm.be/iss</a>	ISO International organisation for Standardization	CEN/ISSS was created in mid-1997 by CEN (European Committee for Standardization) as the focus for its ICT (Information and Communications Technologies) activities.
Cenelec	European Committee for Electrotechnical Standardization <a href="http://www.cenelec.org">http://www.cenelec.org</a>	IEC International Electro technical Commission	Produce the electro-technical standards
ETSI	European Telecommunications Standards Institute <a href="http://www.etsi.org">http://www.etsi.org</a>	ITU International Telecommunication Union	Produce the telecommunications standards that will be used for decades to come throughout Europe and beyond

As the normalisation organizations do not provide fast responses in expected domains, the government agencies in charge of interoperability programmes<sup>38</sup> for data exchange between administration refers to the popular and collaborative de facto standards/Internet standards of the W3C, IETF, OASIS etc.

However, the simple adoption of Internet standards is not enough to cover all the needs. Governments must therefore actively participate to the normalization tasks regarding their specific needs: collect information on best practices in central, regional and local administrations, open discussion forums to officials, business partners and citizens, publish their specific reference documents and syntaxes.

Done on national basis, the results should be consolidated on European level to allow a better interchange of data between the various Member State administrations. To allow a faster adoption than the actual CEN/ISSS norms, the consolidation should be published as a "state of the art" catalogue of best practices, with a 6 monthly or yearly actualisation.

An analysis and recommendation in order to generalize the open source encryption technology (starting from GnuPG for example) may be a good start, as *open source software security platforms* are among the CEN/ISSS preoccupations and as it responds to the European parliament requirements<sup>39</sup>

---

<sup>38</sup> In Italy, the SIU – Sistema Informativo Unitario of the AIPA;  
 In Sweden, the Statskontoret "Government eLink programme concerning e-Government";  
 In United Kingdom. The e-GIF (e-Government Interoperability Framework);  
 In Quebec, the XML deployment strategy expressed in the report "Le XML en route au Gouvernement du Quebec"  
 In France, the "Report Lasserre" on solution interoperability (<http://www.ladocfrancaise.gouv.fr/BRP/notices/004000954.html>) and the already mentioned "Report Carcenac" "Pour une administration électronique citoyenne" (<http://www.ladocfrancaise.gouv.fr/BRP/notices/014000291.html>)  
<sup>39</sup> draft report of the European Parliament's Temporary Committee on the Echelon Interception System, urging the Parliament to encourage future development and greater use of open source encryption technology by business and citizens in Europe.



## Grant interoperable public document access

### Documents

Too often, the electronic documents published in the Internet e-government services by the national and European administration are on proprietary format only.

Out of many examples, the CEN (European committee for standardization) provides a set of electronic templates for the preparation of the European standards, but the CEN-ISO STD automated template (also called "wizard"), known as STD template, can be used only with the Office 97 Microsoft Word version<sup>40</sup>; while the basic templates (with no automated features) must be processed with MS word-processing systems from Word 6 or later...

Such document distribution obliges users to purchase and install a proprietary office suite and are therefore not adapted to an open e-government policy.

The generalized providing of forms in various formats (for example, at least .pdf, .rtf and .doc when text forms are concerned) seems here to be a first step and a minimum (ensuring diversity as a first stage of interoperability).

## Grant interoperable Internet services

The Government web sites are too often using proprietary features that are not compatible to open standards, and may therefore not be accesses by open source software browsers that respect these standards.

---

<sup>40</sup> <http://www.cenorm.be/work/template.htm>

# Legal Issues

## A matter of license

A software licence defines the rights and obligations of both the authors (or owners) of the program – source code and compiled version – and of the user. Without any kind of license, the general public usage (and the modification, enlargement, redistribution) of computer programs would not be admitted according the Bern convention on copyright.

When speaking about specific software from the OSS family, it should help to avoid the use of generic terms as "open source" or "free software".

Although it is difficult for practical reasons, precision is obtained only by referring to the license by name: GPL software, BSD software etc.

Open source doesn't just mean access to the source code, and the Open Source initiative (OSI – Bruce Perens<sup>41</sup>) has established the OSD (Open Source Definition) with nine conditions to comply:

### 1. Free Redistribution

The license shall not restrict any party from selling or giving away the software as a component of an aggregate software distribution containing programs from several different sources. The license shall not require a royalty or other fee for such sale.

**Rationale:** *By constraining the license to require free redistribution, it eliminates the temptation to throw away many long-term gains in order to make a few short-term sales dollars. Without that, there would be lots of pressure for co-operators to defect.*

### 2. Source Code

The program must include source code, and must allow distribution in source code as well as compiled form. Where some form of a product is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost—preferably, downloading via the Internet without charge. The source code must be the preferred form in which a programmer would modify the program. Deliberately obfuscated source code is not allowed. Intermediate forms such as the output of a pre-processor or translator are not allowed.

**Rationale:** *OSI requires access to un-obfuscated source code because you can't evolve programs without modifying them. Since OSI purpose is to make evolution easy, it requires that modification be made easy.*

### 3. Derived Works

---

<sup>41</sup> Bruce Perens wrote the first draft of this document as "The Debian Free Software Guidelines", and it is now a cornerstone of the OSI policy- see at <http://www.opensource.org/docs/definition.html>

The license must allow modifications and derived works, and must allow them to be distributed under the same terms as the license of the original software.

**Rationale:** *The mere ability to read source isn't enough to support independent peer review and rapid evolutionary selection. For rapid evolution to happen, people need to be able to experiment with and redistribute modifications.*

#### **4. Integrity of The Author's Source Code**

The license may restrict source-code from being distributed in modified form only if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. 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.

**Rationale:** *Encouraging lots of improvement is a good thing, but users have a right to know who is responsible for the software they are using. Authors and maintainers have reciprocal right to know what they're being asked to support and protect their reputations.*

*Accordingly, an open-source license **must** guarantee that source be readily available, but **may** require that it be distributed as pristine base sources plus patches. In this way, "unofficial" changes can be made available but readily distinguished from the base source.*

#### **5. No Discrimination Against Persons or Groups**

The license must not discriminate against any person or group of persons.

**Rationale:** *In order to get the maximum benefit from the process, the maximum diversity of persons and groups should be equally eligible to contribute to open sources. Therefore OSI forbids any open-source license from locking anybody out of the process.*

*Some countries, including the United States, have export restrictions for certain types of software. An OSI-conformant license may warn licensees of applicable restrictions and remind them that they are obliged to obey the law; however, it may not incorporate such restrictions itself.*

#### **6. No Discrimination Against Fields of Endeavour**

The license must not restrict anyone from making use of the program in a specific field of endeavour. For example, it may not restrict the program from being used in a business, or from being used for genetic research.

**Rationale:** *The major intention of this clause is to prohibit license traps that prevent open source from being used commercially. OSI want commercial users to join OSS community, not feel excluded from it.*

#### **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.

**Rationale:** *This clause is intended to forbid closing up software by indirect means such as requiring a non-disclosure agreement.*

#### **8. License Must Not Be Specific to a Product**

The rights attached to the program must not depend on the program's being part of a particular software distribution. If the program is extracted from that distribution and used or distributed within the terms of the program's license, all parties to whom the program is redistributed should have the same rights as those that are granted in conjunction with the original software distribution.

**Rationale:** *This clause forecloses another class of possible problem.*

## 9. License Must Not Contaminate Other Software

The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.

**Rationale:** *Distributors of open-source software have the right to make their own choices about their own software.*

*According OSI, the GPL license is conformant with this requirement, as GPLed libraries "contaminate" only software to which they will actively be linked at runtime, not software with which they are merely distributed.*

## A world of diversity

As the listing in the fact sheet (first part of this study) shows there is a great number of licenses and variants. Anyone can create its license (although it should be done with the help of competent lawyers – knowing also some programming)

None of the specific OSS licenses has ever been really tested in court. Are all terms valid? Even if the "client" did not formally sign the license on paper, if it is not written in the user's language etc.? Many of these questions are left open.

The current « OSI certified OSS licenses include (June 2001)

- The [GNU General Public License \(GPL\)](#)
- The [GNU Library or "Lesser" Public License \(LGPL\)](#)
- The [BSD license](#)
- The [MIT license](#)
- The [Artistic license](#)
- The [Mozilla Public License v. 1.0 \(MPL\)](#)
- The [Qt Public License \(QPL\)](#)
- The [IBM Public License](#)
- The [MITRE Collaborative Virtual Workspace License \(CVW License\)](#)
- The [Ricoh Source Code Public License](#)
- The [Python license](#)
- The [zlib/libpng license](#)
- The [Apache Software License](#)
- The [Vovida Software License v. 1.0](#)
- The [Sun Internet Standards Source License \(SISSL\)](#)
- The [Intel Open Source License](#)
- The [Mozilla Public License 1.1 \(MPL 1.1\)](#)
- The [Jabber Open Source License](#)



- The [Nokia Open Source License](#)
- The [Sleepycat License](#)
- The [Nethack General Public License](#)

The first four licenses in that list are the « classical one ».

The GPL is the most used with 80% of the projects in volume, but its viral effect is the most binding: even strongly modified or extended, it is forbidden to redistribute any source code containing GPL on another license than GPL.

The second most used, the BSD, is the most permissive (It is a pure copyright license and derived works can be made proprietary).

Between these two extreme licenses, the more recent MPL Mozilla Public License (released in early 1998) and its variants has since become widely used and is the most convenient to construct an enterprise business model:

The MPL grants the same three essential rights as the other OSI approved licenses (access to source code, right to modify it, right to redistribute it). The production of private derived works is authorised, but modification to the MPL covered code must be made publicly available through the Internet. The MPL is not « viral » in the sense that new developments or extensions (that are not « modifications”) to the MPL code may be distributed with a different license, without publication obligation.

## Evolution toward hybrid or dual licenses

Hybrid licenses like the Apple Public Source License, or the Sun Community Source License are not considered "pure" open source by the older factions of the open source community (although SISSL has been certified by OSI). The licenses still protect the ownership, patent, and other interests of the inventors but publish the source and make it usable to many people. After initial objections, these licenses evolved to become reasonably acceptable to both commercial inventors and the open source community.

Dual licensing is also a common option. One way is to release everything under both a traditional commercial use and an open source license. Users who wish to use the code commercially can provide revenue while providing the benefits of open source to non-commercial and academic users.

The other way dual licensing is done is to lead with the current version under a commercial license, and follow later releasing old, obsolete versions under an open source license. Again many paying users will need the most current version; others can wait for the version to be made open source. Dual licensing seems to be more acceptable to the open source community since it has been done for a longer time.

A license is a contract between Author-Owner and Client-User. As clients tendency is to refuse pure commercial licenses and as enterprise authors are still reluctant to give up core work on pure open source licensing, the natural forecasting is the development of these dual or hybrid license: to reach the objectives of their business model, more and more open source companies will act like traditional companies, and the volume of software distributed under dual or hybrid licenses will grow.

## The Shared source initiative

In line with the above hybrid licenses, Microsoft's shared source initiative is an attempt to create a developer's community, similar to the OSS one. The OSS development model efficiency is here definitely admitted ("given enough eyeballs, all bugs are shallow").

On the key point of transparency, the positive impact of the shared source initiative - if generalized - is that it dissipates the obscurity that was one of the main disadvantages of the classical proprietary software: the black box fear is reduced and security is therefore improved.

On all the other points, the shared source is not to confuse with open source:

- The proprietary character is not modified;
- The relation between developers is not equal (the vendor is the first and financially the sole beneficiary of all improvements);
- Independent modifications and redistributions are not allowed;
- The cost / licensing policy is not modified;
- The source is not equally distributed (possible discrimination).

The shared source initiative will be give transparency and remove indeed one of the main open source advocate's arguments. Concerning a real quality improvement, it is another challenge: It is still to be demonstrated that an important user community will contribute to improve the code in the frame of such relationship.

## Legal coexistence OSS/Proprietary

No proprietary software vendor license ever "forbid" the installation of Open Source Software together with proprietary software. The hypothesis it may occur is not so absurd. If, admitting an advantage of the Open Source development model, a proprietary software vendor decides to open (or to share) its source code to benefit of the user's community active debugging without losing its commercial property rights, this vendor will have reasons to forbid any inclusion of any part its code into Open Source code, since both its proprietary license and the used OSS license may have a "viral" effect.

This should not concern the simple "co-existence" or the pure usage of a commercial application with a GPL one (starting with the popular GPLed Linux, supported by leading commercial actors). This seems now impossible to avoid or to impeach. However, the announced availability of "shared" source code instead of the previously "hidden" code may generate a new range of problems (legal suits) for developers, if it is argued that part of this code was used in GPL product for example, to elaborate filters to convert documents or other interoperability tools.

## The specific GPL controversy



As we discovered in the part 1 of the report (OSS Fact sheet), the GPL is used by about 85% of the projects, although the number of project does not always reflects their importance, as some of the most popular OSS (Apache, Mozilla) are distributed with BSD or other OSI compatible licenses.

The key position of the GPL was highlighted by the recent declarations by one of the Microsoft's senior vice presidents, that – due to its viral character, GPL was a threat to Intellectual property<sup>42</sup>.

*“Companies and investors need to focus on business models that can be sustainable over the long term in the real world economy” and, “the development model, in turn, was almost always based on the importance of intellectual property rights - copyrights, patents or trade secrets.”*  
*“This viral aspect of the GPL poses a threat to the intellectual property of any organization making use of it. It also fundamentally undermines the independent commercial software sector because it effectively makes it impossible to distribute software on a basis where recipients pay for the product rather than just the cost of distribution. In this sense, open source software based on the GPL mirrors the .com business models that proved the least successful during the past year”.*

Another common criticism, this time also from BSD advocates, is that it is difficult to build a business plan around GPL code, and that the BSD licensing model is more business-friendly... The paradox is that so many open source vendors, the major GUI desktops such as GNOME and KDE, the second-generation mail servers such as Postfix and Exim and support organizations (including IBM) choose the GPL-based Linux path instead.

### **Can the viral nature of the GPL damage OSS business?**

It is said that the nature of the GPL weakens the OSS business model, because proprietary and open source approaches cannot be combined in the same business strategy. Challengers of the GPL like to spread fear about its "viral" effect, but what is it exactly? Just one thing: if you include or link at runtime GPL code in your own application, it should become<sup>43</sup> GPL also. In all other situations, for example if proprietary and GPL code just interacts through API's or if proprietary applications or drivers make normal calls to a GPL operating system, there is no impact.

In fact, almost all software licenses have the same "viral" effect: for example, if you obtain and include Microsoft code into a derivate, this one should normally belong to Microsoft.

---

<sup>42</sup> Craig Mundie, Microsoft Senior Vice President “The Commercial Software Model” Lecture at The New York University Stern School of Business - May 3, 2001

<sup>43</sup> “it should” because the GPL and this viral effect has never been tested on Court.





The GPL does not require a company to give away source code of its custom application programs: a vendor can distribute commercial software with commercial licenses together with GPL software.

This is the reason why the GPL licence is considered as full compliant to the 9<sup>th</sup> OSI condition that stipulates: “**License Must Not Contaminate Other Software**”. This means that the license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software, because Distributors and end-users of open-source software must keep the right to make their own choices about their own software.

The GPL is conformant with this requirement, as GPLed libraries "contaminate" only software to which they will actively be linked at runtime, not software with which they are merely distributed.<sup>44</sup>

A vendor will have good reasons to avoid GPL software if he wants to include it in its original proprietary solutions, keeping some improvements secret to the operating system - for example, network communication or video decompression software. In the embedded world, unlike in the raw server world, such changes are made at a deep level, making it difficult to keep the public software separate from the proprietary software. The GPL that covers Linux doesn't allow to keep this deep level mixing proprietary and that is the reason why a proprietary vendors like Wind River Systems<sup>45</sup> has decided to deliver its solutions with a BSD licensed embedded operating system.

In other case of embedding where the operating software is just a commodity, or for selling dedicated servers for example, where the property of the operating system is not the core business, the inclusion of a GPL software like GNU/Linux provides the best value without negative effects: most of the new low-cost dedicated servers are now Intel/Linux (3Com internet server, Superstack 3 webcache, Right Vision Eye Box One and Eye Box Pro, Sun Cobalt CacheraQ4, RaQ4, Staquare...)

Now what about the idea that GPL software is "not sustainable over the long term in the real-world economy" and that the open-source business model is "linked to failed dot-com start-ups"? Companies that have acquired a strong position in licensed software, but are weaker or non-existing in integration and services generally launch the idea. It is contradicted by embedded Linux companies and by major constructors as IBM, HP, Sony, Sharp, Ericsson, Nokia, Motorola, Samsung and other moving from licensing to services, that have based a substantial part of their economic development on the GPL software model.

---

<sup>44</sup> <http://www.opensource.org/docs/definition.html>

<sup>45</sup> Wind River acquired BSDi, the owner of the BSD/OS, the version of Unix at the heart of open-source projects such as FreeBSD, OpenBSD, and NetBSD. Under the terms of its license, though, anyone may keep modifications secret.



# The Liability due to OSS delivery

## Principles

What if a public sector agency distributes – for free – software out of its own production?

Software can produce two categories of damages:

- Direct damages (if information is erased or damaged, its replacement, restoration, intervention of technician, time directly related to the error correction and to the reconstruction of lost data, etc.);
- Indirect damage (loss of activities, loss of image, poor performances due to IT trouble and loss of confidence towards the IT system, cost of time not directly related to the error, but related to the finding of an alternative solution, the obligation to migrate again all data to the new solution, the training of the personal to another product, etc.).

The open source character (and transparency) of the software is not a guarantee against damages, since most of users do not open the source and cannot understand all implications of native code. In addition the most “transparent” program may operate perfectly until unforeseen environment, parameter, conditions, involuntary or even intentional misuse could be harmful to persons, properties or information.

Without a license agreement attached, a government agency may (theoretically) be made fully liable for any consequences (direct or indirect) of bugs and malfunctions of the delivered program.

The first indication is therefore to avoid distribution of software without a clear license that excludes the liability for these direct or indirect damages. As all Open Source Software licenses contain such disposition, GIGA estimates that there is neither special liability nor protection from liability in using an Open Source License<sup>46</sup>. For example, the GPL contains this paragraph:

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

---

<sup>46</sup> IdeaByte . No Special Liability nor Protection From Liability in Using an Open Source License  
RIB-112000-00029 - 2000 - Giga Information Group

Similar exclusion clauses are found in other OSS licenses, and on this specific point, it is significant to remark that proprietary licenses (as for example, the Microsoft EULA license) include very similar or identical wording. The inclusion of such clauses appears then as an “automatic reflex” and, although mainly due to the consumer protection law the validity of these clauses is questionable, OSS licenses does not present here significant differences with others.

## Validity of the license agreement

The second recommendation is to ensure that a “valid License contract” has been concluded between the distributor and the licensee. In the case of downloading, the simple fact the license is published “somewhere on the same public internet site” (often written in English when the user speaks German, French or Italian) will not be considered as a sufficient evidence of a valid contract between this user and the distributor.

Very soon, the generalisation of electronic authenticated signature will bring a convenient solution to this problem by permitting the request of a formal and proven user’s approbation of the license conditions before any (even free) downloading. In the meantime, the “hand written” signature on the license agreement stays the best guarantee.

## Gift or for sale – Virus liability

If the software is sold, (which is the case of most popular packaged distribution) the vendor should consider the various laws protecting consumers: these laws impose minimal obligations (and right to be reimbursed) during minimal periods, and all contrary dispositions, even agreed by the consumer are void.

If, On the contrary the Open Source Software is given for free, downloaded “as a gift”, the question of “reimbursement” is obviously non relevant, but - even with a license excluding any kind of liability, this will not protect the distributor against complaints if the software is “damaging by nature” as it is easily possible if the software contains a virus.

The liability is here dependent of the “level of fault”, that can be intentional, gross negligence, simple negligence, or liability by endangering without any fault, even in case of involuntary introduction of the damaging virus or “*force majeure*”.

## Distribution by whom? To whom?

An issue related to the potential liability is the identification of the “responsible vendor”, and here the OSS model presents particularities: in the case of free downloading of a cooperative open source software, any kind of evidence will be difficult to provide. The source code itself – if downloaded – can provide more evidence, with its copyright indications and annotations.

If the author or distributor is clearly identified, its “quality” (public sector agency or commercial vendor) is generally reinforcing the liability: IT professionals are supposed to act with competence and experience, according to the state of the art in their branch. Private end-users or even non-IT commercial enterprises will benefit (in Court) from a reinforced protection.

Such case of liability for Open Source distribution has never been tested in court (and thus, some will argue that the “legal scarecrow” is perhaps more a paper dragon than a real one). However, the core function of most public sector agencies is not to distribute software, except if it is closely related to their role (E. g. an income declaration checking software, distributed by the competent tax administration).

Therefore, if an administration wants to license software (e.g. according to one of the OSS licensing models), it should better not licence it directly to end-users, but – on a non exclusive base – to “First level licensees” (a dedicated competent public agency that will check the inoffensive character of the software or/and a panel of specialised OSS vendors, association or user’s groups which will organise distribution and downloading). In order to grant a free utilisation without the risk that its software become proprietary of distributed with a more restrictive license, a GPL type license, accepted and signed by the candidate, in which the candidate endorses possible liability towards end-users, is the solution. If at the contrary the administration intention is to permit also the inclusion of the software in proprietary products (without exclusivity), then a MPL type licence is recommended.

In addition, this (GPL, MPL type or other) license must specify both:

- The applicable law (European directives and a specific Member State law). There are minor differences between the Member States, but serious differences between European and American law (also concerning other points, as software patents), and applicability of American or other extra-European law should be avoided under any circumstances;
- The competent court (located in a Member State) in case of any contestation. The location of the competent court, E.g. “Brussels” or “Berlin” may be different from the origin of the applicable law, E.g. “Belgian” or “German”.

Especially if dealing with private persons, the determination of the applicable law and of the location of the court is not always legally binding or valid. To avoid any possible contestation of the above points, it is recommended that the “First level licensee” (dedicated public agency or commercial bodies) should be clearly located in the European Union.

## **Market and fair competition**

### **§ 81 of the EC Treaty**



Paragraph 1 of the new Art 81 (ex article 85) of the treaty establishing the European Community (Consolidated versions incorporating the changes made by the Treaty of Amsterdam, signed on 2 October 1997) prohibits “*agreements between undertakings, decisions by associations of undertakings and concerted practices which:*

...  
“*(d) apply dissimilar conditions to equivalent transactions with other trading parties, thereby placing them at a competitive disadvantage;*”

The above article creates also constraints for governments (according the new Article 86, “*to public undertakings and undertakings to which Member States grant special or exclusive rights*”) and as the public sector generates a substantial part (15%) of the global IT market, it should then avoid “*concerted practice*” which apply “*dissimilar conditions to equivalent transactions with trading parties*”. Is it the case if, for example call for tenders systematically impose the use of specific proprietary commercial software (although competitors, including open source, exist)?

As Article 81 paragraph 2 stipulates that “*any agreements or decisions prohibited pursuant to this Article shall be automatically void*” governments should carefully avoid any form of exclusions resulting from proprietary software pre-selections, if it is not strictly imposed by interoperability with existing infrastructure.

According to Article 81, paragraph 3, the provisions of paragraph 1 may, however, be declared inapplicable in the case of a practice which:

“*contributes to improving the production or distribution of goods or to promoting technical or economic progress, while allowing consumers a fair share of the resulting benefit, and which does not:*

- (a) impose on the undertakings concerned restrictions, which are not indispensable to the attainment of these objectives;*
- (b) afford such undertakings the possibility of eliminating competition in respect of a substantial part of the products in question”.*

These derogations are not applicable if the Terms of Reference impose an already dominant product, but what if it imposes the use of “open source software”?<sup>47</sup>

In this case, the requirement of the use of OSS should be justified according Article 81 paragraph 3 (promotion of technical or economic progress, allowing consumers a share of the resulting benefits) – for example with an explicit reference to the e-Europe initiative, parliament recommendations or similar motivation.

In addition the notion of OSS should be clarified in such call for tenders: indeed, because a simple reference to “open source” or “free software” creates confusion. A clear reference to the license by name (GPL, BSD, MPL, etc.) should be the most precise formulation, but may not be practicable because a large variety of licenses exist and because it should create discrimination. Therefore, it is recommended to refer explicitly to the OSI conditions or – better - to specify which of the 9 OSI conditions<sup>48</sup> if some of them (or all of them) are really required and justified. For example, if the main

---

<sup>47</sup> This occurs recently in several calls for tenders. For example, the EU/ IDA 2001 Circa Architecture (DG Entr/01/48) which stipulates as general requirement “all components of the future architecture should be based on Open Source Software – following the expressed goals of both EIONET and e-Commission / e-Europe initiatives”

<sup>48</sup> see at <http://www.opensource.org/docs/definition.html>

requirement is “security through transparency” and therefore the sole condition concerns the providing of the source code without needs for redistributing it, the benefit of an hybrid license as the Apple PSL, the SUN SCSL or the Microsoft “Shared Source” licence will fulfil the condition.

The same remark is to be done concerning “open standards”: the term is too vague and it should be recommended to enumerate clearly which standard framework (from which official of “de facto” organisation”) is required.

## § 82 of EEC treaty

Article 82 (ex Article 86) of the EEC treaty (revised by Amsterdam treaty) stipulates that:

*“Any abuse by one or more undertakings of a dominant position within the common market or in a substantial part of it shall be prohibited as incompatible with the common market insofar as it may affect trade between Member States.”*

Such abuse may, in particular, consist in:

...

*(c) applying dissimilar conditions to equivalent transactions with other trading parties, thereby placing them at a competitive disadvantage;*

Two hypotheses may here be formulated:

a) The case of term of reference requiring OSS

If the conditions are clearly formulated by a reference to a published list of requirements or to a published list of standards, this does not introduce any dissimilar conditions, provide the requirements are precise and really correspond to a need (as in the above example, if the furniture of the full source code is the sole condition, it should be excessive to generally require “Open Source Software” or the compliance to all the 9 OSI conditions)

b) The case of European Union or Member States licensing OSS products for free

This may also be seen under certain circumstances as unfair competition according to this article 82. As the core business of European Union or of Governments is not to license software, the possible impact on a “substantial part” of the common market must be evaluated prior to act, depending on the purpose of the software and of the existence of commercial actors on that specific market.

If the software may be considered as a commodity for all citizens, a GPL type licensing should be recommended – under non discriminative conditions as those reported above concerning the license policy regarding liability – through agencies, groups or developer associations acting naturally according the OSS model (in Germany, the Government helped the German Unix User Group to release the GNUpg encryption product according to a GPL license).

If the software may be considered as usable in industry (also by proprietary vendors), a MPL type license is more recommended as it allows the production of private derived works that may be distributed with a different license (modifications of MPL covered code must be made publicly available through the Internet).

A dual licensing policy (GPL, or MPL for commercial redistribution) is also possible.

The European Court of Justice has not until now considered a case in this area, but if the government (or European Union) purpose corresponds to their core objectives (promote e-Government for citizens and enterprises, and facilitate interchange of data between administrations) and if they act on a neutral way according to this objective, without discriminating any commercial actor, we estimate that no one can complain regarding “*abuse by one or more undertakings (or Member State, or Public enterprise) of a dominant position*”.

## § 87 of EEC treaty

Article 87 (ex Article 92) of the EEC treaty (revised by Amsterdam treaty) stipulates that “Any aid granted by a Member State or through State resources in any form whatsoever which distorts or threatens to distort competition by favouring certain undertakings or the production of certain goods shall, insofar as it affects trade between Member States, be incompatible with the common market.

But paragraph 3 allows to provide two kinds of aids:

(a) aid to promote the execution of an important project of common European interest or to remedy a serious disturbance in the economy of a Member State;

(b) aid to facilitate the development of certain economic activities or of certain economic areas, where such aid does not adversely affect trading conditions to an extent contrary to the common interest;

These exceptions allow Member States to promote projects, even with commercial purpose (via the national funds for scientific research or other public organisations). Although, the European Union itself is not formally concerned by the text of article 86, this should also concern the European administration and programmes, as the 5<sup>th</sup> IST framework programme supporting OSS projects (development of certain economic activities).

A general distribution of specific software dedicated to the extension of e-Government (e.g. in the field of signature authentication or privacy) is also compatible to the European rules according to paragraph 3 a) “important project of European interest”.

## The Software Patent question



The original purpose of Intellectual Property is to protect authors against illegal copying of their works and innovators against competitors: those who have not invested years of time and money in creation, research and development are not authorized to copy and resell innovator's findings.

Intellectual Property includes two distinct branches: Copyright and Patent. A third way to protect industrial assets exists: Trade secrets that are also extensively used in software industry, by non-publication of the source code and of interchange format.

The OSS developers do not contest Intellectual Property, quite the contrary. If a piece of code is taken "as is" or included into a new solution, the author's name cannot be unfairly "removed" or the license conditions ignored. The whole OSS licensing policy, and the GPL in particular, is really based on intellectual property: copyright.

As it is the case for literature of music, a copyright system protects the software authors. It may be extended to the Internet distribution, with a payment per download agreement for authors requiring it, for free if authors grant it, and this would not fundamentally restraint the freedom to innovate or to elaborate collaborative works.

The situation of OSS developers regarding a possible extension of patent law to software is not different nor in opposition to other developers, but they are certainly more sensitive to possible problems, just because they understand better the principles of copyright law, patent and the notion of infringement.

Patent Law in general, provides a radically different protection: indeed, even if a developer starts from a real "clean room" (without creating a derivative work of someone else's copyrighted material, without ever stealing/studying/being inspired by someone else's work) the 10 lines computer program that he or she wrote this morning might have infringed some patent.

Software-related patents can lead to the award of very significant damages, as it was proven in the US by several cases as *Stac Electronics v. Microsoft Corp.*, No. C-93-0413-ER (C.D. Cal.), where the parties ended up signing a broad cross-licensing agreement in the spring of 1994 after a federal jury in Los Angeles awarded Stac \$120 million for patent infringement and Microsoft \$13.6 million for trade secret misuse.

## The Law

In Europe, the matter is regulated by the European Patent Convention (EPC)<sup>49</sup> and in particular by its art. 52 that defines the three conditions of "patentability"

### Article 52 - Patentable inventions

- (1) European patents shall be granted for any inventions which are susceptible of industrial application, which are new and which involve an inventive step.
- (2) The following in particular shall not be regarded as inventions within the meaning of paragraph 1:
  - a) discoveries, scientific theories and mathematical methods;
  - b) aesthetic creations;

---

<sup>49</sup> The full text is published by EPO at <http://www.european-patent-office.org/legal/epc/e>





- c) schemes, rules and methods for performing mental acts, playing games or doing business, and **programs for computers**;
  - (d) presentations of information.
- ...

According to art. 52, (2), c) the actual patent law does not concern computer programs.

## The reality

In recent years, the United States Patent and Trademark Office (USPTO) has granted a rapidly increasing number of patents for software-related inventions. By some counts, software patents now number in the thousands, and as many as 10,000 applications for such patents are pending<sup>50</sup>.

In a kind of reaction, the European Patent Office (EPO) in Munich has also delivered a great number of software patents. As there is no explicit identification for software related patents, an estimation of their number can result from certain keywords retrieval like internet, server, client or virtual into a database. It is not easy to determine if these keywords belongs to the core subject of the patent or are just given incidentally. Depending on estimation, the number of EPO software related patents goes from 10.000 to 30.000

Another way of estimation is to calculate the number of patent owned by companies involved in the software industry: Here is the "top ten" <sup>51</sup>

Rank	Company	Number of Patents
1	IBM (US)	1842
2	CANON KK (JP)	1364
3	SIEMENS AG (DE)	1104
4	SONY CORP (JP)	1069
5	NIPPON ELECTRIC CO (JP)	874
6	FUJITSU LTD (JP)	742
7	TOKYO SHIBAURA ELECTRIC CO (JP)	706
8	MATSUSHITA ELECTRIC IND CO LTD (JP)	626
9	HEWLETT PACKARD CO (US)	504
10	HITACHI LTD (JP)	495

Globally this number of patent was estimated to more than 38.000 (and more than 16.500 for the top 50 enterprises). Of course, all these patents are not related to pure software. A good part of them concerns hardware. Another part concerns compound industrial products including software and hardware. The specific software patent part should not exceed 40% of the total.

<sup>50</sup> 10000 in the US according the SPI Software Patent Institute - <http://www.spi.org/primintr.htm>, and 12550 in the US according the project "Software Patent Statistics" <http://swpat.ffii.org>

<sup>51</sup> For the "top 50 list" see [http://swpat.ffii.org/vreji/pikta/perled/app\\_stat.html](http://swpat.ffii.org/vreji/pikta/perled/app_stat.html)





The conclusion of our estimation and a comparison with the US situation is that EPO software patent applications should be about 9.000 (so, reasonably less than the 20.000 to 30.000 estimated by the FFII, Federation for a Free Informational Infrastructure<sup>52</sup>)

## The Controversy

The current trend towards extending the scope of intellectual property (and software patents in particular) constitutes a major threat against Open Source software, since the free software community (individual developers, public sector and education, SMEs) is under equipped to defend itself in a world where considerable financial resources are required for patent portfolios or infringement litigation.

Furthermore, as a consequence of the openness of its source code (nothing is hidden), free software is particularly vulnerable to infringement suits. The main threat concerns patents on interoperability: communication protocols and formats.

As software technology has continued to advance, using combinations of previous advances in order to create new goods and services is a natural process. These advances have caused uncertainty and controversy about ownership and transfer of the rights to use hybrid software-related technology.

Because most software technology has traditionally been held as a trade secret or published in non-patent publications, patent examiners do not have ready access to such technology in trying to evaluate the patentability of a software-related invention disclosed in a patent application. They have often proven to be unable to estimate the “triviality” or the pre-existence of intellectual methods or algorithms located in software. In light of this controversy, the League for Programming Freedom and others have called for the abolition of patents for software-related technologies.

Why should software development benefit of a special regime?

The entire notion of “prior art” is a kind of nonsense when applied to software development. Few developers start from a blank page: they usually start from various pieces of existing code and routines, some of them coming from applications having no relations or common need with the new software purpose (and just serving as guideline, reference, skeleton, or providing sub routines that may be re-used). And even when starting from a blank screen, the developer focuses on the result. He tries also to keep the code clean, well organized and consistent, or even “elegant” but has no time and no technical means to permanently check if he does infringe some patent. This is the main reason why a wide application of patent law to code writing should constitute a substantial hurdle to software development.

A recent MIT study illustrated that for industries like software, there is actually good reasons to believe that software “incremental imitation” **promotes** innovation, and that strong patents (long patents of broad scope) **inhibits** it<sup>53</sup>.

---

<sup>52</sup> FFII site: <http://www.ffii.org/>: « European Patent Office has been granting against the letter and spirit of the written law for the last few years. For reasons explained below, it is difficult to find out the precise number, but it seems the EPO must have granted at least 20-30,000 software patents”

<sup>53</sup> James Bessen and Eric Maskin, “Sequential innovation, Patents and imitation” – MIT January 2000 No 00-01- p. 2

Applied to software, many patents are not used to protect industrial invention but as an aggressive way to lock IT sectors against competition. In addition, it gives to IT giants the resource to register more and more patents, that they may after exchange (or cross license) with other IT giants, while individual developers or small enterprise cannot deploy the same costly strategy.

For developers that are writing code just like political columnists are writing press articles or man of letters are writing books, software patents represent a major threat concerning a fundamental liberty. Their fear is that, as the number of software patents grows exponentially, it will soon become impossible to write any program without a serious risk of falling foul of some patent.

## The EPO demand

To ensure that software patents are made legal, the European Patent Office expects a revision of the European Patent Convention (EPC) that actually stipulates (article 52, 2, c) that "programs for computers... .. shall not be regarded as inventions".

Several legal institutes<sup>54</sup>, patent specialists or Industry representatives share this view, and the EPO demand received support from experts as Prof. Lutterbeck of Berlin Technical University, his assistant Robert Gehring and Axel Horns, patent lawyer in Munich, figuring under the name of "Internet Governance Research Group" (that received an order from the German Ministry of Economics and Technology to work out a "short expert opinion" which was published in December 2000). The report states that Art 52 EPC was a misconception from the beginning, and that patent law will be seriously impaired unless any innovation that is implemented through a computer is patentable. However the report warns "*software patents can have a very negative impact on open source software and proposes that patent law at least in Germany should be amended in such a way that the publication and transmission of source code does not violate the law, even if the execution of object code on a computer does.*"<sup>55</sup>

---

<sup>54</sup>E. g. the Max Planck Institute for International Patent Copyright and Competition Law in Munich - <http://swpat.ffii.org/vreji/papri/eukonsult00/planck/indexde.html>

<sup>55</sup> <http://swpat.ffii.org/vreji/papri/bmwi-luhoge00/indexen.html>

The EPO request is also motivated by the preoccupation to avoid different policy in Europe and in the US, that should damage the position of European software industry and create discrepancies in a global economy. If a product infringes a US patent, it can (and often is) blocked from being imported into the country. For example, the US customs may block any boxed Linux distributions containing infringing code. Obviously US customs will not be able to stop users downloading through the Internet, but stopping boxes would cut enough revenues to prevent the inclusion of such code in the boxes. The vendor location in Europe (E.g. Mandrake in France or SuSE in Germany) is not important, because it is the place where the product is exported and distributed that is important: all software distributions are subject to US Industrial Property law when selling/exporting products to the US. The consequence is that to keep economical consistency, the software patent question cannot be solved at European level only, and must be negotiated and co-ordinated between the EU, the US and other concerned countries

## The reactions

The software patentability concerns three of the European Commission directorates and two legislative organs (the European Parliament and the Council of Ministers). The decision process was therefore slow enough to initiate numerous reactions, mainly from the OSS community with the relay of parliament and government members.

In October 2000, the European Commission Industrial property unit announced<sup>56</sup> the opening of an official consultation on the economic and social impact of software patents in Europe and more than 100 motivated papers were collected. The European Commission's consultation organisers have started to publish a part of the submissions since January 2001<sup>57</sup>. Meanwhile the FFII<sup>58</sup> has prepared a systematic overview and evaluation of the EC consultation replies<sup>59</sup>.

Originally, the consultation rather supported the viewpoint of the European Patent Office and was accompanied by a study in that direction. Most of patent law experts and professional associations approved the extension of patentability, but at the same time OSS users associations, companies and about 1000 individual developers expressed concerns and opposition to the patentability of software, business methods, intellectual methods and other immaterial objects.

The Eurolinux association launched a petition<sup>60</sup> for a Software Patent Free Europe and says it is now supported by more than 80.000 signatories<sup>61</sup>, 2.000 chief responsible officers of IT enterprises and 300 companies.

---

<sup>56</sup> See on [http://www.europa.eu.int/comm/internal\\_market/en/intprop/indprop/softpaten.htm](http://www.europa.eu.int/comm/internal_market/en/intprop/indprop/softpaten.htm)

<sup>57</sup> See on [http://europa.eu.int/comm/internal\\_market/en/intprop/indprop/softreplies.htm](http://europa.eu.int/comm/internal_market/en/intprop/indprop/softreplies.htm)

<sup>58</sup> Federation for a Free Informational Infrastructure

<sup>59</sup> See on <http://swpat.ffii.org/vreji/papri/eukonsult00/>

<sup>60</sup> See on <http://petition.eurolinux.org/>

<sup>61</sup> It would then be the largest cyber-right signature campaign in the Internet so far, as the second largest – directed against spamming – received about 30.000 signatures

At governmental level, the German Minister of Justice, Ms Däubler-Gmelin, explained<sup>62</sup> in a newspaper interview of May 3rd her opposition to plans of the Directorate of the Internal Market to codify the practise of the European Patent Office (EPO) and legalise software patents by means of a European directive. The French minister of economics, Christian Pierret had already expressed a similar opposition before, and the British "e-minister" Patricia Hewitt had also expressed reserves. The Dutch parliament had decided that the problem of triviality check must be solved before a legalisation of software patents can be envisaged. While the patent department of the German ministry of justice is avoiding comments and discussions, the ministry of economics has ordered two studies from patent-friendly sources, but after an expert hearing in the German Parliament<sup>63</sup> it seems that the consensus of most German political fractions against software patents was rather reinforced.

The consultation was interrupted and its results published only partially. The European Commission's Industrial Property Unit will revise its initiative and present a new project at the end of 2001.

---

<sup>62</sup> See on <http://www3.computer-zeitung.de/cz/archiv/artikel/artikel.988808215.20124.html>

<sup>63</sup> <http://swpat.ffii.org/penmi/bundestag-2001/indexen.html>



# Recommendations

## Information Collection

It is still difficult to report on the use of open source, because most of European countries just ignore them in government policies (Italy, Sweden) or do not isolate them in statistics (Spain). France and Germany are still at the beginning of their information campaign to make public sector aware of the possible benefits of open source

It is important that government start to monitor the (sometimes hidden) use of OSS in their administration. The purpose of this data collection is twofold: inform, and get information back. The regular organisation of “software best practice days” – giving speaking time to OSS advocates – is one of the concrete actions to undertake.

## Reflection on OSS benefit of ownership

Together with the information collection, the public sector should lead a reflection on the Total Cost of Ownership / Benefit of ownership models and determine the different cost structure obtained, thanks to the scaling effect, with both the proprietary and the Open source model.

The impact on the quality and motivation of the team (from passive user to active member of a development community), the importance of transparency, open standards, independence, should also be evaluated.

Public sector IT budgets and terms of references should be more neutral and open to the alternative OSS model “from licensing to services” allowing the transfer of resources from software license acquisition to integration and human training

## Respect of standards

It is often reported that certain Public sector Web sites use proprietary non-compatible standards. The use of a specific proprietary web-browser (even distributed for free) is declared “mandatory”. This should be avoided as it creates or increases discrimination. Similarly, attached documents are proposed only in proprietary format .DOC of read-only .PDF, and open interoperable versions (e.g. RTF) are missing.

## Shared investment in new standards

Governments should actively invest in creating and sharing a wide set of adapted XML data structures. Some of them are initiating the work (e.g. France). But the danger is to create new e-borders by adopting “national” data structures. In order to obtain an interoperable European frame work for al kind of administrative documents, the governments should themselves adopt what is working the best with OSS: the development model. The European union role may be to propose the resources of a common mediator or facilitation web host (a kind of XML data structure Source Forge) with the collaboration of standardisation groups.

Objectives are avoiding administrative complications and re-inventing the wheel; publishing immediately and sharing free of charge all new data structure. One of the challenges of this European shared XML data standardisation platform would be to respect the diversity of languages and cultures: all government or public sector agencies should be able to participate on a peer to peer level.

## **Certification authorities**

As the conformity to “open standards” (and perhaps to the “Open Source” definition) will be more and more required, how to evaluate a vendor declaration that his software is “in conformity to the standards”. To avoid endless contestation, public (and private) sector will need published frameworks of standards, or list of conditions, and independent from vendor authorities that will certify the software and license compliance to a determined level of standardization or open character.

## **Call for tender**

The public sector's calls for tender and terms of reference have also to be “open” to OSS. Without going to the other extreme (e.g. imposing GPL software) they should at least be neutral and avoid imposing – without real necessity regarding user's needs – any kind of specific proprietary software.

On the contrary, mentioning that the solution must be “Open source” or “In conformity to the open standards” is unclear and should be avoided: the conditions to satisfy should be indicated clearly and they should be motivated by real user's needs.

## **Promotion of a global transparent market**

When purchasing a new PC in a shop, the client will notice that hardware and software are not separated but bundled together. With the exception of Macintosh, the client has usually no other choice than a “Windows + Intel (wintel)” platform. If the client really wants to install Linux for example, he has to remove Windows (or create a partition on top of it) and may loose support and guaranty from the vendor until the original configuration is restored (and all his work gone).

Without any break in neutrality, it should be possible to agree with constructors and large PC distribution shops to provide alternative solutions reflecting the cost of bundled software.

## **A clear limit to patentability**

Any revision of the European Patent Convention must clearly differentiate software from inventions with an industrial application that can be patented normally, even if software is included in the patent, as “closely associate to the industrial process of producing a specific material good”. On the other hand, the pure intellectual, logical processes, or mathematical algorithms should not be patented, unless – as a compromise - for a short duration, if it requires to protect their innovative character by allowing a company to launch its new software, and provided a “non triviality” examination has been done.

\* \*  
\*

