



Free and Open Source Software - a feasibility study

Appendix 1: Extensive survey

This publication is available from:
The Swedish Agency for Public Management
Publication Service
Box 2280
SE-103 17 Stockholm

Phone +46 8 454 46 43
Fax +46 8 454 46 45
E-mail: publikations.service@statskontoret.se
www.statskontoret.se

© STATSKONTORET
ISBN: 91-7220-526-1
Redners tryckeri, 2003

Content

1	INTRODUCTION.....	7
1.1	FREE SOFTWARE AND OPEN SOURCE SOFTWARE.....	7
1.2	SHORT BACKGROUND.....	7
1.3	CONCEPTS.....	9
1.4	LICENSES.....	15
1.5	DEVELOPMENT AND MAINTENANCE.....	18
2	TECHNOLOGY FORECASTING.....	23
2.1	CURRENT SITUATION WITHIN THE EUROPEAN UNION.....	23
2.2	EUROPEAN INITIATIVES – EXAMPLES.....	26
2.3	INTERNATIONAL INITIATIVES – EXAMPLES.....	32
3	PRESENT SITUATION – PUBLIC ADMINISTRATION IN SWEDEN.....	40
3.1	SMHI – SWEDISH METEOROLOGICAL AND HYDROLOGICAL INSTITUTE.....	40
3.2	SWEDISH PREMIUM PENSION AUTHORITY (PPM).....	41
3.3	SWEDISH NATIONAL LABOUR MARKET ADMINISTRATION (AMS).....	42
3.4	MUNICIPALITIES.....	43
3.5	MALMO GENERAL HOSPITAL (MAS).....	43
3.6	LINUX IN THE SCHOOLS.....	43
3.7	CURRENT ENVIRONMENT IN PUBLIC ADMINISTRATION.....	44
3.8	PROJECTS AND INVESTIGATIONS WITHIN SWEDISH PUBLIC ADMINISTRATION.....	44
4	WHY FREE AND OPEN SOFTWARE.....	45
4.1	”WHY OPEN SOURCE SOFTWARE” – DAVID A. WHEELER.....	45
4.2	THE ADVANTAGES OF FOSS PRODUCTS.....	46
4.3	WORKSHOP ON AREAS OF USE.....	47
4.4	SUCCESSFUL FOSS PROJECTS.....	52
4.5	SECURITY ASPECTS.....	57
4.6	THE PUBLIC PERSPECTIVE.....	58
5	OPEN FORMATS AND STANDARDS.....	59
5.1	OPEN FILE FORMATS FOR OFFICE SOFTWARE.....	60
5.2	INTEROPERABILITY WITH OTHER PRODUCTS.....	61
6	THE LOCK-IN EFFECTS OF VENDOR DEPENDENCE.....	63
6.1	THE MECHANISMS OF LOCK-IN.....	63
6.2	HOW TO DEAL WITH LOCK-IN.....	65
7	USING FREE AND OPEN SOURCE SOFTWARE.....	69
7.1	THE MARKET FOR FOSS IN SWEDEN.....	69

7.2	FINDING THE RIGHT PRODUCT	70
7.3	CAN FOSS AND PROPRIETARY SOFTWARE BE COMBINED?.....	70
7.4	MIGRATING DESKTOP COMPUTERS	71
7.5	NEED FOR SERVICES	72
8	FINANCIAL AND LEGAL ASPECTS	74
8.1	BUSINESS MODELS FOR FOSS	74
8.2	COST COMPARISONS	78
8.3	LEGAL ASPECTS	87
9	SUGGESTIONS FOR FURTHER WORK.....	92
9.1	COMPETITION RELATED MEASURES	92
9.2	CATALOGUE OF PRODUCTS FOR THE ADMINISTRATION.....	95
9.3	MEASURES TO IMPROVE KNOWLEDGE	96
9.4	PILOT INSTALLATIONS	98
9.5	INVESTIGATE IN-HOUSE DEVELOPMENT OF FOSS	99
9.6	LEGAL ASPECTS	100
10	REFERENCES.....	102
10.1	BIBLIOGRAPHY.....	102
10.2	OTHER LITERATURE	105

1 Introduction

1.1 Free software and open source software

There are two main types of software that are considered "open" and which are covered in this report. They are *free software* and *open source software*. Free software is what the name implies. The software is freely accessible and can be freely used, changed, improved, copied and distributed by all who wish to do so.

Although open source software is just as "free" it does not have to be free of charge. Besides being able to construct business models around the software based on commercial aspects, a company can receive direct payment by the use of a large number of licensing schemes and models, e.g. concerning support, further development, etc. What is important here is that the source code is freely available to the customer.

In this report we choose to treat both free software and open source software in the same category, hence the abbreviation FOSS, which is used here.

For proponents of FOSS, openness and unexclusiveness are fundamentally a question of democracy and a basic tenet is that information should be freely accessible to everyone and all without any restrictions. This does not mean that everything on the Internet should be free of charge. But what it does mean is that the inherent architecture of the Internet should be free.

1.2 Short background

UNIX was introduced in 1969 by AT&T Laboratories. It was more or less free to use in the academic world. Since the major part of the further development of Internet technologies took place within a university and research environment, UNIX became the operating system of choice for this development. The sharing of source code among software developers became commonplace.

At the beginning of the 1980's AT&T changed its licensing conditions and started charging a fee for all use of UNIX. As a result of this, other computer manufacturers such as IBM, HP and Digital Equipment (DEC) started to develop their own proprietary UNIX-based operating systems.

At that time, Richard M Stallman worked with software development at the Massachusetts Institute of Technology (MIT). He started work at the MIT laboratory for artificial intelligence in 1971 and quickly became a part of a group of programmers who exchanged ideas and source code with each

other. In 1980, computer manufacturer DEC stopped the development of a type of computer, which MIT had been using. The alternatives at hand all used proprietary operating systems, which were not free to use and definitely not free to distribute or pass around. On the contrary, one had to sign a non-disclosure agreement in order to be able to even install these operating systems. The knowledge one obtained concerning an operating system could not be made public or made known to anyone else. The sharing of program code among members of the group ceased to take place.

Stallman chose to resign at MIT in 1984. Instead he started the project called GNU (GNU's Not UNIX). Stallman wanted to recreate the environment he had experienced in the development group at MIT, but on a larger scale. He wanted to create a free operating system. He felt he had to quit at MIT in order not to be affected by any requirements or demands concerning any copyright to his work.

In addition, he created a special kind of license called GNU (General Public License) in order to guarantee that any software developed within the GNU project would be free and available for anyone to use. In 1985 Stallman founded the Free Software Foundation (FSF) for obtaining financial support for the GNU project.

The licensing model differs much from other software licenses, mainly concerning the purpose of the licensing agreement and the scope of what the license covers. GNU GPL is based on the freedom to make agreements concerning licensing conditions and the premise that the originator of the software has the right to make the software available on his or her own conditions.

GNU GPL became the object of wide attention, but not everyone was attracted to the puristic and ideological spirit in which Stallman worked. Soon, there were calls for a system of free distribution of source code without having to be connected or be affiliated to the philosophy, which Stallman had created and manifested in GNU GPL.

Bruce Perens developed a set of principles called "The Debian Free Software Guidelines" aimed at dealing with software development. The Open Source Initiative (OSI) was founded in 1997 by, among others, Bruce Perens and Eric S. Raymond. They came up with an outline of open source code called "The Open Source Definition" based on the above-named software guidelines. All licenses, which conform to the definition, can be called "Open Source Software". The major difference between this definition and GNU GPL is partly that commercial sales of software are allowed and also that "Copyleft" (see below) is not mandatory. However, it should be noted that GNU GPL could be placed in the category of FOSS (free and open software).

1.3 Concepts

Commercial software, available without access to source code and thereby impossible to change, improve or to be redistributed, is called proprietary software. The existence of licensing conditions and stipulations for proprietary software, with limits as far as use is concerned, is the main reason for the development of FOSS. Licenses for proprietary software generally only give the user the right to use the software under certain conditions without any knowledge of how the software is designed and without any access to its source code.

Within the area of free and open source software there are two main camps: those who look upon free software as a social and moral issue and who see proprietary software as a social problem, i.e. those who subscribe to the tenets of the Free Software Foundation (FSF), and those who agree more with the Open Source Initiative (OSI), which represents a more pragmatic attitude and who see FOSS as an efficient model for software development

The difference between the two is more than just a name, but instead a question of two fundamental value-based orientations. It is important to keep the two concepts apart when discussing philosophies and values, but also to understand that the two concepts do not counterbalance each other, but instead work together in the advocating of free and open source software.

The figure below, which can be found on FSF's homepage, shows how the different types of software relate to each other.

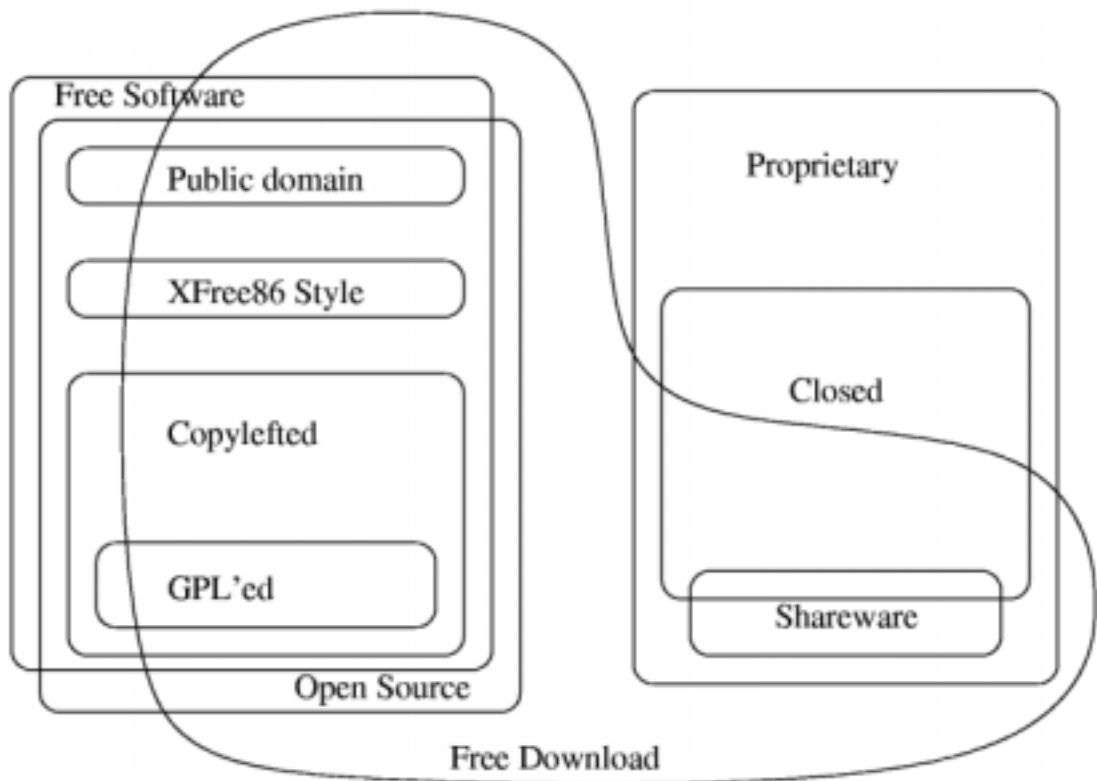


Figure 1: How various types of software relate to each other

1.3.1 Free software – Free Software Foundation (FSF)

According to FSF free software is a matter of the user's freedom to run, copy, distribute, study, change and improve the software. More precisely, it refers to four kinds of freedom, for the users of the software:

0. The freedom to run the program, for any purpose (freedom 0).
1. The freedom to study how the program works, and adapt it to your needs (freedom 1). *Access to the source code is a precondition for this.*
2. The freedom to redistribute copies so you can help your neighbour (freedom 2).
3. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). *Access to the source code is a precondition for this.*

The complete definition of free software, "The Free Software Definition" can be found at: <http://www.gnu.org/philosophy/free-sw.html>.

A program is free software if users have all of these freedoms. Well-known licenses, which comply with the definition, are GPL and LGPL.

1.3.2 Open source software - Open Source Initiative (OSI)

Open source doesn't just mean access to the source code. The distribution terms of open source software must comply with the Open Source Definition by OSI, which is defined by the following criteria:

1. Free redistribution without royalty requirements.
2. The program must include source code or there must be a well-publicised means of obtaining the source code.
3. The license must allow modifications and derived works.
4. Integrity of the author's source code.
5. The license must not discriminate against any person or group of persons.
6. No discrimination against fields of endeavour.
7. The rights attached to the program must apply to all to whom the program is redistributed.
8. The license must not be specific to a product.
9. The license must not place restrictions on other software that is distributed along with the licensed software.
10. The license must be technology-neutral.

The complete definition of open source software according to the OSI can be found at <http://www.opensource.org/docs/definition.php>.

Software, which is distributed with any OSI, certified license can be called "OSI Certified Open Source Software" and is entitled to use the OSI certification mark.

Well-known licenses, which comply with the definition, are e.g. GPL, LGPL, BSD, MIT and MPL (see section 1.4 Licenses). At the present time (January 2003) there are 43 different certified licenses.

1.3.3 GNU

The name GNU is a recursive abbreviation of "GNU's Not Unix". The logotype shows a gnu, an animal living in freedom. The GNU project, which was initiated by Richard M. Stallman, has as its objective to create a system which is compatible, but not identical to, UNIX. The GNU project is, however, not limited to only an operating system. The project offers a wide range of software, including applications.

The GNU project consists of many different subprojects, which are supported by voluntary expertise and also by companies whose aim is to create a component within the GNU system.

The GNU system is, much like other UNIX systems, of a modular construction, and especially GNU systems with Linux as a core – GNU/Linux systems – are widespread and are part of most Linux distributions.

1.3.4 Copyleft

Copyleft is a mechanism, in among others GPL licenses, which stipulates that free software remains free, even when modified or changed. Copyleft does not allow anyone who redistributes the software, in original or modified form, to add any restrictions to the license. This means that all copies of the software, even modified, continue to be free.

Copyleft stipulates also that all who redistribute the software, with or without changes, must see to that the copyleft function remains intact in all distributed copies. Copyleft guarantees in this manner that the software continues to be free for all users.

An example of a license with copyleft is GNU GPL.

1.3.5 Free software without Copyleft

Free software without Copyleft allows the user to add restrictions to new distributions of the software. If software can be deemed free, but without copyleft restrictions, modified copies can be distributed with limitations and without right of use, etc. A commercial vendor can then distribute the software – with or without modifications – as proprietary software.

Examples of licenses without copyleft are BSD (Berkeley Software Distribution) and MIT (Massachusetts Institute of Technology).

1.3.6 Proprietary software

Proprietary software is not open. Use is regulated and further distribution and modification is either forbidden or demands special permission by the supplier or vendor. Source code is normally not available.

1.3.7 Freeware

The term "freeware" has no clear definition, but is generally used for software, which allows copying and further distribution, but not modification and where the source code is not available. Freeware should not be mistaken for open software.

Freeware is distributed in binary form (ready to run) without any licensing fee. In some instances the right to use the software is limited to certain types of users, for example for private and non-commercial purposes. Freeware is often used in marketing situations in order to sell additional products and to gain market advantages. One example is Microsoft Internet Explorer, which is made available as freeware.

1.3.8 Shareware

Shareware is software, which is made available with the right to redistribute copies, but it is stipulated that if one intends to use the software, often after a certain period of time, then a license fee should be paid.

Shareware is not the same thing as free and open source software (FOSS) for two main reasons. The source code is not available and modifications to the software are not allowed.

The objective of shareware is to make the software available to try for as many users as possible. This is done in order to increase prospective users' will to pay for the software. The software is distributed in binary form and often includes a built-in timed mechanism, which usually limits functionality after a trial period of usually one to three months.

1.3.9 Commercial software

Commercial software is the term used for software, which is developed by an organisation or company with the purpose of making a profit. Commercial software and proprietary software is not the same thing!

The majority of all commercial software is also proprietary, but there are also examples of commercial open software, where the supplier hopes to make a profit on added value services and supplementary products.

1.3.10 Public domain software

Public domain software is software, which is not copyright protected.

Placing software in the public domain means, in reality, that all claims to copyright are relinquished. Public domain software exists – in principle – only within certain areas in certain countries, e.g. the United States. In the US large quantities of public domain software is produced in a university environment with the support of public authorities. This software is available to all without any restrictions. But it is generally not allowed to use the software for commercial purposes.

1.3.11 Shared Source

Shared source is a relatively new policy from Microsoft. Its purpose is to give certain groups (public authorities and administrations) access to peruse source code within a Microsoft product. One has to sign a contract, which regulates what one can and cannot do with the information obtained pertaining to the source code.

Shared source is not the same thing as FOSS, because the fundamental rights to copy, modify and distribute are lacking.

1.3.12 Open formats and standards

In order to achieve an increase in competition and interoperability between products without “customer lock-in”, file formats, for example in word processing, must be based on open standards. It is first then that users can achieve concrete possibilities to freely choose between products, a choice that can be based on needs and prices.

If document exchange or file formats together with exchange of information between products can be standardised, products can compete with each other as far as functionality and quality are concerned, without the present-day locking in of the customers.

The dominating actors on the market will probably not base their products on open standards without continued strong pressure from consumers who require them. Open standards benefit consumers by providing the prerequisites for competition with reduced prices and increased quality.

One example of an open source project with the ambition of creating an open XML based file format is the *XML project*. This is a project under the auspices of OpenOffice.org with the aim of creating an open file format for office documents and for creating an open reference for this format. The project states that it will work for standardisation between products for office support.

1.4 Licenses

One of the main reasons for attaching licensing conditions to FOSS products is in order to protect applicable rights and freedoms and to guarantee that the software remains open, even in the future.

Licensing conditions can among other things:

- Safeguard defined rights
- Safeguard conditions, which the copyright owner wishes to protect (naming of the development team, distribution of license with the software, etc)
- Safeguard the rights of the copyright owner
- Ensure that modified software based on FOSS remains open
- Ensure that all software, which uses FOSS in a system, is also open

Licenses for FOSS vary the most when it comes to the last two points above. While the GPL license is very strict when it states that all software based on the original source code shall remain open and based on the same license, the BSD license allows use of the source code for the development of proprietary software. It is the principle of copyleft (see section 1.3, Concepts), which separates GPL from many other licenses for FOSS.

On the other hand, most licenses for FOSS do not present any obstacle for a company, which wishes to sell commercial products based on FOSS. But those who sell products based on FOSS cannot, on the other hand, prevent others from doing the same thing. Licensing conditions do not allow any limitations to the right to copy and change the software. Also, source code must be included in the distribution.

It must be made clear that products based on FOSS are FOSS license regulated.

OSI has formed a definition for open source software, see previous section 1.3, Concepts. Licenses, which meet the demands of the definition, are approved by OSI, and products, which are registered under any of these licenses, have the right to be called *OSI Certified Open Source Software*. At the present time (January 2002) there are 43 certified licenses. See <http://www.opensource.org/licenses/>.

Restrictions and rights in licenses for FOSS must be considered in the light of each other in order to maintain both the author's intentions and the needs of the end-user. Because the purpose and idea of restrictions and rights vary, this has resulted in the development of a large number of different licenses

for FOSS. In the following section we give a brief introduction to some of more common licenses for FOSS.

1.4.1 GNU General Public License (GPL)

GNU GPL is today the most common license for free software. The most well known examples are products from the GNU project and Linux. GPL was created by Richard M. Stallman and exemplifies in principle, the philosophy of the Free Software Foundation (FSF)

There are no restrictions concerning copying and distribution, but certain conditions must be met. Source code must be easily accessible for the user, the license must be included with the distributed software and screens must be shown stating these conditions upon starting the software.

Modifications are allowed as long as changes are clearly stated, when modified source code is written and by whom. Derivative works are allowed, but must be published using the GPL license (the copyleft principle). What is known as the viral effect comes from the distribution of software products, containing GPL licensed software, which must also be licensed using a GPL compatible license. In that way, software licensed with another license, will be “infected” by the GPL license. This intentional effect means that GPL is not suitable at all in commercial products with proprietary software, because in order to take advantage of the GPL licensed source code as far as integration or inclusion is concerned, one must also open up one’s own proprietary software.

The GPL license has not been tested in any extensive judicial or legal context, either in Sweden or in the rest of the world. It can therefore be speculative trying to interpret how certain conditions would be assessed in a court of law. A user accepts the conditions of the license as soon as the software is run. Infractions to the conditions can, among other things, mean a loss of the right to use the software, which, in turn, can clash with the Swedish Contracts Act §36 concerning adjusted conditions and negligible infringements, i.e. a much too extensive effect in relation to the extent of the crime. The GPL license also does not take into consideration non-commercial law. Work is currently being carried out, within FSF Europe, with the purpose of looking at how GPL can be adapted to European legal conditions.

The GNU GPL license is described in its entirety at <http://www.fsf.org/licenses/gpl.html>.

1.4.2 GNU Lesser General Public License (LGPL)

Since GNU GPL is restricted for commercial use, FSF has developed an alternative license called Lesser General Public License – LGPL.

The LGPL license allows commercial or proprietary software to use parts of a source code, e.g. libraries, without the viral effect of GPL. One of the reasons for FSF publishing LGPL was that free software of a “better” quality ran the risk of too low a rate of acceptance and distribution in order to become a de facto standard and to be able to compete with proprietary products. One of the first products to be licensed with LGPL was the GNU C libraries.

At the present time, however, FSF recommends use of the GPL licenses rather than the LGPL.

Url: <http://www.fsf.org/copyleft/lesser.html>.

1.4.3 Mozilla Public License (MPL)

Mozilla Public License (MPL) is the name of the license for Netscape’s source code for the Mozilla browser. Mozilla is the open source software, which the Netscape browser is now based on. MPL contains a variation of *copyleft*, which allows MPL licensed software to be incorporated in other software and to be published without the risk of any viral effect of the copyleft principle. Therefore, one might say that the construction of MPL is similar to LGPL. Other similar licenses are IBM Public License and Sun Public License.

Url: <http://www.opensource.org/licenses/mozilla1.1.php>.

1.4.4 MIT License and BSD License

The BSD license gives all users the right to make copies of the software, including documentation, without cost. It also gives the user the right to use the product for commercial purposes. The user has the right to copy, modify, integrate, publish, distribute, re-license under other licenses and sell copies of the software without any other restrictions than having to inform of these rights, along with the name of the owner of the original copyright.

Url: <http://www.opensource.org/licenses/mit-license.php>
<http://www.opensource.org/licenses/bsd-license.php>.

1.4.5 Comparison between different types of licenses

The following is a summary, taken from the FLOSS report, of different types of licenses. More information on the legal aspects of the licenses can be found in section 9.6, Legal aspects.

Table 1: Comparison between different types of licenses (FLOSS, part 3)

Type of license	Free of charge	Further distribution permissible	No restrictions for use	Source code available without cost	Changes in source code permissible	Derivative work must also be open	Integration with proprietary software permissible
Public domain	X	X	X	X	X		X
Shareware	(X) ¹	X					
Freeware	X	X	X				
GPL	X	X	X	X	X	X	
LGPL	X	X	X	X	X	X	X
MPL	X	X	X	X	X	X	X
BSD License	X	X	X	X	X		X

1) Shareware is only free for a limited period of time for evaluation of the software.

1.5 Development and maintenance

Boston Consulting Group has presented results from studies on how FOSS is developed, who takes part in this development and their motives. One observation is that there is generally a high degree of creativity in development projects for FOSS. Another observation is that those who participate in the projects are, for the most part, professional developers, who take part, either as a part of their work, with the permission of their employer or on their own time.

Characteristic for the participants is their strong feeling of identification with *the community*. Intellectual challenge and possibilities for developing their talents are also important driving forces. To “be seen” is also important and this contributes to the will to achieve good results. In a closed environment, where an individual programmer is not always observed in the same way, commitment can sag and the result can sometimes be mediocre source code.

A typical community consists of a core group of committed and knowledgeable individuals - sometimes chosen or elected - and sometimes

on their own initiative - who keeps a focus on the work and chooses the best contributions. Anyone - both companies and individuals - can submit a contribution, but it is the core group who decides on issues such as system architecture, etc. and which contributions should be used. If an individual doesn't deal with his "assignment" in a professional manner he can be voted off the group, or a completely new group can take over, so-called *forking*. Linus Torvalds is a good example of an extremely successful leader of a community – that for the development of Linux.

The community is responsible therefore for development, further development and maintenance of the product. By and large, all such projects set up their own Internet portal where those who are interested can follow the software development, read the documentation, download new releases, both stable releases for use and versions under development for purposes of testing, voice opinions and contribute to the project itself.

1.5.1 Cathedral and bazaar models

There is a wealth of literature written on development models for software. The description of the two above-named models is taken from Eric S. Raymond's book "The Cathedral and the Bazaar" (see 10.2, Other literature)

The cathedral model is drawn upon centralised planning and execution and describes the traditional way of developing proprietary software. Development is strictly hierarchically organised according to a top-down model of centralised control. The name comes from the medieval way of building cathedrals.

Adversely, the bazaar model is built on decentralised planning and execution and describes generally the manner in which FOSS has been developed. Work is organised in networks. A community of developers works with the development and improvement of optional components and parts of the software. Most commonly, the Internet and a joint central web server are used to collect and publish the material. The name comes from oriental bazaars where anyone can offer his goods for sale and anyone can visit the bazaar.

A model, which goes somewhat further than the bazaar model is the cooking pot model. At a bazaar, goods are exchanged, but their functionality or purpose of use is not. In a "cooking pot" single ingredients can be mixed with each other, resulting in something new and of a greater value than the basic sum of the ingredients.

In reality, there are, of course, mixtures of all three models. FOSS is often developed according to the cathedral model in its first stage. Then an early version of the software is published, often workable, but unfinished. This is then further developed according to the bazaar model and the cooking pot model, where users can add modules and give the software new functionality.

A number of generally decisive success factors can be distinguished towards the end of the development cycle:

- Functional requirements must be met; i.e. the software must work as planned.
- Software quality must be sufficiently high.
- The software is constructed modularly; it must be easy to add and change modules.
- Development time (time to finished product) must be short so that adjustment to new hardware can be expedient. (Within certain areas this factor is of less critical importance.)
- As low development costs as possible.

As we have seen, different development models and processes for proprietary and FOSS are used. Central control is used in the development of proprietary software for determining what functions should be implemented. In the model for FOSS, development takes place on the participants' own conditions and at their own initiative. As an individual, one has the opportunity to suggest and produce changes and additions. However, a hierarchical model is used even in the development of FOSS with a core of individuals, enthusiasts, who decide which components are to be included in the finished product. Compare Linus Torvalds role in the development of Linux.

It is often said that the ironing out of bugs and errors is a quicker process in the development model for FOSS because there are generally more individuals who review and correct errors in the software. Hypothetically, one could then expect – calculated as a percentage – a fewer number of errors in FOSS!

1.5.2 Driving forces

Financial driving forces, which normally are pointed out as necessary for the development of proprietary software, are almost totally lacking for FOSS, i.e. reimbursement for and profitability of those resources needed for the development of the software.

What it is that drives developers to develop FOSS is a complex matter. More than half of all software developers who develop FOSS are professional developers, employed by a commercial player with its own interests in making FOSS a successful venture. For example, one can mention the sales of servers with Linux as an operating system and the sales of supplementary software and services.

What type of FOSS has the preconditions for being a serious competitor to proprietary software? Must it be a general application with a reasonably great market base? And can FOSS exist without its counterpart: proprietary software?

Various studies have shown that it is the **competence-related motives**, which weigh the most among developers of FOSS. They want to learn from each other by contributing themselves and by learning from the knowledge of other developers. Among the professional developers who develop both proprietary and FOSS, the latter bazaar and cooking pot models are preferred where more space is given to individual creativity. The time factor connected with proprietary software development with scheduled deadlines is not the same, and a developer gets quick feedback on his or her proposals.

Besides the above-named competence related motives, the following driving forces can be seen as decisive factors:

Career-related motives: Many developers want to make a good show in the face of competition, for among other reasons in order to create a better foundation for getting a well-paid job.

Personal motives: Many developers enjoy creative programming and are fascinated by software development as such. They like being originaive and many are dedicated to their own, personal work. A comparison can be made to writing music, an avocation, which occupies many people, who do it without any ambitions of making money on their interest.

Ideological motives: Many developers support Richard M Stallman's philosophy that it is a democratic freedom that all software should be free. Many other developers generally dislike the commercial interests, which control the range of software products on the market.

Financial motives: There is a motive among many to be able to make money on a business idea based on FOSS. Companies such as HP, IBM, RedHat, Sun, etc, are in this category.

Need-related motives: Sometimes there simply is no software, which fits the bill. This is mainly the case within the fields of research and education. New software is developed which is then released as FOSS.

1.5.3 Developers

Developers of FOSS come from a range of different environments. The prevalent areas are universities and colleges, public authorities, companies and individuals.

Universities and colleges produce a large amount of software for reasons of education and research. Even though a part of this software becomes proprietary, most of it is published as FOSS. Many talented students contribute also to the development of this software, partly because they like programming and can learn from other programmers and partly because it can have a beneficial effect on their own careers.

A number of distributors of FOSS also take an active part in the development by contributing themselves by enhancing or making the product better. Their motive is most often either a demand from their own customers or even making the product better on their own and thereby increasing their profit.

Besides distributors, even commercial software companies with their own interest in FOSS, take part in its development. Examples are IBM, HP and Sun who set aside considerable resources for the development of FOSS. These three companies also market the Linux operating system and therefore have a strong interest in contributing to its development.

2 Technology forecasting

2.1 Current situation within the European Union

2.1.1 IDA (Interchange of Data between Administrations)

[IDA](#), Interchange of Data between Administrations, a program under the auspices of the European Commission, arranged during the latter part of February 2001, under the guidance of Erkki Liikanen, a symposium in Brussels on the topic of free and open software in the public sector - [Open Source Software in EU public administrations - February 2001](#). The symposium came to the following conclusions:

- There is extensive experience in the use of free and open software in the public sector in Europe
- Current projects include both server installations (back office) and client computers in the workplace
- Free and open source software is used because of adaptable functionality, lower overall costs, vendor independence and adherence to open standards, interoperability and security.

A number of measures were suggested:

- Upon procuring IT services, more importance should be placed on the use of open standards, including standards for document formats and exchange.
- Authorities should supply information on possible solutions based on free and open software.
- A forum for the exchange of best practice in the use of free and open source software should be established
- A software pool, i.e. a model for enabling exchange of software developed by members, should be established.

Furthermore, it was established that there is a need for efforts to outline both cost and benefits of free and open source software. Also the legal aspects need to be covered.

As a result of this symposium IDA published a report – [”Study into the Use of Open Source Software in the Public Sector - June 2001”](#) - with information on FOSS and a description of the situation at that time (2001) within the European Union. The report contains three parts: an informative section (OSS fact sheet), a part which describes both use and experience of FOSS (OSS usage and experience) and a part which gives an overview of the market for FOSS (2001) and informs of procurement aspects.

In June 2002 another report was published – ["Pooling Open Source Software \(POSS\) Feasibility Study - June 2002"](#) – which describes a possible model for how to manage and exchange in-house developed software with other users in the European Union.

In order to create "best practice" in this area, IDA started a pilot project in the autumn of 2002. The project comprises a pre-study prior to a possible migration and introduction of a solution based on FOSS for authorities in the German regional state of Mecklenburg-Vorpommern. The project will gather and present information on costs and difficulties arising when migrating from a proprietary to an open computer environment. The solution aims at comprising both servers and around 15,000 workplace computers, which have SuSE Linux as an operating system.

A study group with Swedish representation will follow up the project. The project is expected to continue for about two years, but the results from the first stage will be presented during the spring of 2003.

A further proposal from the IDA covers a study into the preconditions for establishing a centre of competence for FOSS. Furthermore, there are proposals for carrying out a study of open formats for document exchange.

2.1.2 IST – Information Society Technology

In 1999 a study group was formed, established at the initiative of the Information Society Technology Directorate General within the European Community, with the purpose of analysing the area of free software (Libre Software) and make recommendations for continued work. The recommendations are generally the same as those presented at the IDA symposium in Brussels (see previous section). The report - [Information Society Opportunities for Europe – April 2000](#) – was presented for the first time at the IST '99 conference in Helsinki, Finland.

The IST has a homepage with information on activities, initiatives and projects within the EU - [Free / Open source software actions in European programmes](#). At this time there are almost twenty on-going projects within the Fifth European Community Framework Programme, with direct concentration on the usage of FOSS. Projects can be found in the areas of PKI and security, information management, health care, education and also an interesting study on the socio-economic aspects of FOSS. There is also information about the [strategy document](#), which the Information Society Technologies Advisory Group (ISTAG) has published, with recommendations for the sixth Framework Programme (2003-2006). The group recommends among other things an increased use of FOSS and usage

of the development models according to open source: *"Where appropriate and supporting the general policy, the use of Open Source licensing should be stimulated for software generated in the IST Programme, and indeed in all parts of the sixth Framework Programme"*.

2.1.3 The FLOSS project (IST)

The socio-economic project FLOSS (Free/Libre Open Source Software) presented an interesting report in June 2002, entitled [Free/Libre and Open Source Software: Survey and Study](#).

One of the project's purposes is to provide for the demand for information on FOSS. The report starts with an historical background and then presents, using a number of studies, a clear picture of both usage and development. Concurrently, indicators have been identified for measuring the added-value effects and spreading of FOSS. The report covers also business models and the effects, which recommendations for FOSS can entail. Finally, an assessment is made of what these recommendations can result in as far as social and economic effects are concerned.

The report is divided into several parts along with a summary covering use, companies, public sector and the EU, basic information, development and an overview of software on the market.

Studies were made of, among other things, the degree of usage of FOSS in four areas within companies and authorities in England, Sweden and Germany: operating systems, databases, websites and workplace computers.

Most common is the use of FOSS as an operating system, e.g. Linux and Free/OpenBSD at an average of about 16%.

The use of FOSS for databases was about 11%. Examples of such software are MySQL, PostgreSQL and Interbase.

Next comes use of FOSS for creating and setting up websites. About 10% use the surveyed software such as Apache, PHP, Perl, Python and Squid in their web servers. The report states that there are probably a large number of hidden or unidentified users of FOSS in this area, where Apache, for example, has a 65% market share of all web servers on the Internet.

The use of FOSS such as Linux, Mozilla and OpenOffice/StarOffice in workplace computers only has a penetration of about 7%.

2.1.4 e-Europe 2005: An information society for all

e-Europe is an initiative, which was introduced by the European Commission during 1999 for the purpose of encouraging the development of IT in Europe. An action plan for the initiative was created and named *e-Europe 2002*. This plan is now superseded by the action plan *e-Europe 2005*. One of the suggestions is to create a framework for interoperability in order to support e-services to individuals and companies in Europe. "*e-Europe 2005 interoperability framework will be based on open standards and encourage the use of open source software.*"

The action plan for e-Europe conveys the message that interoperability is important and that co-ordination must take place on a national level. The development of a framework for interoperability is a current activity within the Commission.

2.2 European initiatives – examples

FOSS has achieved a solid penetration in Europe. Initiatives for the promotion of distribution and use of FOSS within public administration are now taking place on both on a national and on a EU-level.

2.2.1 The Nordic countries

Public administration in all Nordic countries is involved in questions surrounding FOSS. The current market situation with dominating vendors and inadequate competition is the same in all the countries.

2.2.1.1 Denmark

Work is being carried out at the present time in Denmark for arriving at a strategy for how FOSS should be used in public administration.

The Danish Board of Technology presented a report in October 2002 entitled "[Open Source Software in the Digital Public Administration](#)." The report points out the fact that Danish public administration can save up to €480 million per year within as short a time span as four years by changing over from proprietary software to FOSS. In a further perspective the potential for saving is €730 million per year. See more in section 8.2.2, Report from the Danish Board of Technology. Savings can be made in the first place in the areas of workplace computers, office software, operating systems and business-related systems in combination with an extended lifetime for PCs and servers.

The report contains a description of FOSS. A study is made of FOSS for workplace computers, in an infrastructure, in solutions for web systems and in business-specific systems. The report also contains detailed financial analyses concerning the use of FOSS within these areas along with a study of the financial consequences this use can involve pertaining to public administration in Denmark.

The report states that FOSS should be considered a serious alternative to proprietary software within Danish public administration. The report recommends that the Danish federal government and other authorities formulate principles and goals for the procurement of software. The principles should include the introduction of open standards, an important precondition for achieving better competition where products based on FOSS can serve as an alternative.

As a phase in making FOSS available and at arriving at cost-cutting effects, the educational authorities in Denmark have arrived at an agreement with Sun Microsystems resulting in free use of StarOffice. The Department of Education requires that all free access to software shall be distributed via UNI-C, Denmark's IT-centre for research and education.

2.2.1.2 Finland

The Ministry of Finance along with other government agencies in Finland have evaluated OpenOffice. Both Windows and Linux have been used as operating systems in the evaluations. Continued work is being done to arrive at guidelines for national authorities concerning the use of FOSS in their specific information systems.

In the city of Turku, a project has been carried out in order to ascertain whether or not Linux and OpenOffice can replace Microsoft Windows and Office. Among other reasons for starting the project was the announcement by Microsoft that licensing terms and conditions were to be changed.

The evaluation showed that OpenOffice could not yet provide a match for MS Office as far as functionality was concerned, but that the product worked well and the functions that were available were more than needed for most users.

The advantages noted concerning Linux as an operating system were stability, security and also the fact that Linux has not been affected or has been as vulnerable as the MS-environment has to virus attacks.

The Turku municipal IT department recommended therefore a transition to Linux and OpenOffice. A pilot installation was initiated in a real time working environment in November 2002. After an evaluation of functionality and costs, the total migration will commence at the beginning of 2003.

2.2.1.3 Iceland

Iceland is a small language area and is strongly dependent on individual vendors. One example can be found in the leading office software on the market. This software is not localised for the Icelandic market.

Recently, the Ministry of Justice in Iceland decided to use StarOffice at all police stations in the country, a total of about 700 PCs.

2.2.1.4 Norway

In Norway, the unaffiliated agency, *Statskonsult*, published a report on FOSS as early as 2001. *Statskonsult* is organised under the Department of Employment and Administration in Norway. The report points out considerable advantages of FOSS as far as cost savings and increased competition are concerned. Among other suggestions there are the following measures that can be taken in public administration:

- Encourage use of Linux and FOSS within school and educational systems. The motive given is that students and teachers should acquaint themselves with several different products. Furthermore, money can be saved on licensing costs.
- Support the development of FOSS for creating alternatives to current proprietary software. FOSS can produce positive competition on the market by putting pressure on vendors of proprietary software to develop and adapt their products. FOSS can also contribute to a reduced dependency on individual vendors.
- Evaluate the possibility of publishing in-house developed and copyrighted source code as FOSS.

During the summer of 2002, the joint agency called FNS (*Forvaltningsnettsamarbeidet*) (under the Department of Employment and Administration, the Norwegian Association of Local Authorities and the Agency for Public Administration) decided not to renew its core agreement with Microsoft. At this time the Ministry of Employment together with the Ministry of Industry and Commerce is working on an ICT strategy for the

public sector and in conjunction with this strategy an evaluation is being made of the possibilities of increasing the use of FOSS. The IT strategy will be published, according to earlier reports, in February 2003.

As in the case of Iceland, there are problems with the fact that only a few software products are localised and translated to both *Nynorsk* and *Bokmål*, the two official Norwegian languages.¹ However, three municipalities have commissioned a translation of OpenOffice, and they now have local distribution of Linux for use in the schools.

2.2.2 Great Britain

In July 2002 a report was presented in Great Britain concerning a policy for FOSS:

- "UK Government will consider OSS solutions alongside proprietary ones on IT procurements. Contracts will be awarded on a value for money basis."
- UK Government will only use products for interoperability that support open standards and specifications in all future developments."
- UK Government will see to avoid lock-in to proprietary products and services."
- UK Government will consider obtaining full rights to bespoke software code or customisations of COTS (Commercial off the Shelf) software it procures wherever this achieves best value for money."
- UK Government will explore further the possibilities of using OSS as the default exploitation route for Government R&D software."

In procurement situation the solution that presents the best financial value will be chosen. This can either be a product based on FOSS, a proprietary product or a combination of them both. The choice can vary depending on the conditions involved.

It is important to pay attention to the retaining of interoperability between systems. This is especially important when launching e-services.

¹ There is at the present time also an effort called the Linux for Schools project, which organises the translation of a number of Linux programs to Bokmål, Nynorsk and Sami (the Lap language of northern Norway).

An endeavour should always be made to lower costs and risks for government systems. This can be done by:

- purchasing best value for money solutions
- removing the reliance on individual IT vendors
- providing more flexibility in the development, enhancement and integration of systems
- vesting the ownership of bespoke and tailored software code with the government

Security is an extremely vital area. FOSS products can be considered just as secure as proprietary products and at the current time are less susceptible to attacks via the Internet (virus, etc). In certain cases proprietary "standard" products are considerably less secure than the corresponding FOSS alternatives.

In order to follow-up on the above-named policy:

- The OGC (Office of Government Commerce) will update their Procurement Guidelines to reflect this policy
- Advice will be made available to all those involved in procurement exercises on areas of the software infrastructure and application marketplace where OSS has strengths and weaknesses.
- Advice will also be made available to all those involved in procurement exercises on how to assess the merits of OSS versus proprietary solutions in procurements.
- OeE (Office of the e-Envoy) and DTI (Department of Trade and Industry) will discuss with academic research institutions the possibilities of future R&D work

URLs: www.govtalk.gov.uk, www.e-envoy.gov.uk

2.2.2.1 Police forces in England

Police forces in England (UK Police IT Organisation) have signed a contract for delivery of Linux-based desktop computers. The first delivery is to the West Yorkshire force with an installed base of 3,500 computers. If the pilot scheme is a success, continued delivery of the Linux-based computers could cover over 60,000 desktops.

For West Yorkshire alone, there is a calculated projected savings of £1 million a year (€1,4 million).

The system solution is based on a relatively simple hardware base with Linux as an operation system, OpenOffice as office software and an application for thin clients where a user connects to the business end of the

system and receives basically screens from the other system. This means that a user with a Linux-based computer can run server-based Microsoft applications.

All computers are stateless and a user can log on with their smart card from any machine and have access to their authentication and user profiles.

2.2.3 Spain – Extremadura

In the west Spanish region of Extremadura with 1.1 million inhabitants, the regional government decided in April 2002 to initiate a campaign for migrating all of the region's computer systems from Windows to Linux.

In October 2002, about 10,000 desktop computers had been migrated and it is calculated that about 100,000 more will have changed over by the end of 2003.

Luis Millan Vazquez de Miguel, a one-time college professor who changed over to a career in politics heading the area of education and research, says: "We are the future. If Microsoft doesn't become more open and generous with its code, people will stop using it and it will disappear."

During an interim period, many users will need to use both Windows and Linux in order to be able to communicate with others and other systems, but one can hope that this initiative will reach out, both within Spain and other countries in the EU.

Authorities in Extremadura have engaged a local supplier for arriving at a package solution containing both operating system and office software. The name given to this particular distribution is "Linex", a contraction of the words Linux and Extremadura. The distribution has become so popular, that over 55,000 downloads from www.linex.org have been registered to people living outside the region.

The regional government has also invested in a competence and development centre for developing business-adapted systems for finance and health. These systems will also be distributed free of charge.

Over 150,000 CDs containing the produced systems have been distributed to schools, electronics shops, municipal offices and as inserts in magazines. Even TV has been used to market the production. Computer vendors in the region have reached an agreement to pre-install the software package instead of MS Windows.

2.2.4 Germany

In Germany, the Ministry of the Interior has decided to use Linux in public administration, and it has signed a special agreement with IBM to this effect. There is interest in not becoming dependent on any single vendor. An important reason is also to make the infrastructure more secure by avoiding what one calls a "monoculture". The agreement with IBM can be used both by central administration, the regional governments and by municipalities.

Schwabish Hall, a city of 35,000 inhabitants, has decided – with the agreement as a focal point - on implementing a completely Linux-based infrastructure. Plans are in the making for using Linux both for servers and for the desktop environment and for using OpenOffice as office software. An initial 120 desktop computers will be installed with the new systems with a gradual increase to about 400. There are also plans in the making for creating a competence centre for Linux in the local public administration.

The Ministry of the Interior has also issued a report on open file formats: "Open Standard for Document Exchange", published in August 2002. The purpose of the report is to stimulate debate by showing different ways of developing and maintaining an open standard for document exchange.

2.2.5 France

In France, there are various activities being carried out within public administration for promoting the use of FOSS. ATICA (Agence pour les Technologies de l'Information et de la Communication dans l'Administration) has a forum for disseminating knowledge of FOSS. There is a great need for information on options for training and support, and examples of such information are now available. The agency also organises training sessions for public administration.

In a recently published report from ATICA, "Guide de choix et d'usage des licences de logiciel libre pour les administrations" (Guide for administrations on selection and use of licenses for FOSS), advice is given on which licenses public administrations should use in the development of FOSS, and that an administration should make a careful analysis of its needs and licenses. The report recommends that GPL be used in the development of software. In that way one achieves the best chances of distributing and protecting developed software.

2.3 International initiatives – examples

At the current time, there are at least 20 different countries all over the world where governments have taken a favourable position on the use of FOSS. In some of the countries, such as Peru and other countries in South America, the governments have chosen legislation as a way to promote use of FOSS. Other countries have chosen a less strict approach.

The most interesting areas right now are India and China with their enormous populations. These countries can influence the future course and direction of the software market. Both countries have a very positive view concerning the use of FOSS, mostly for stimulating local business initiatives and for minimising the outflow of currency, but a striving for national independence also plays an important role. Undoubtedly, the largest growth for FOSS during 2003 will take place in Asia.

Several research and advisory companies say that the strong growth shown in the IT sector in China, is due to a consistent course of direction concerning and also the use of FOSS. The Chinese government has supported locally produced software, based on Linux, mainly for financial reasons. But security aspects have also played an important role, because of the demand for control in the development of software. The government in Taiwan has officially demanded that Microsoft release source code for this same reason.

The Chinese company, Redflag Software, founded by the Chinese Academy of Sciences, produce a locally adapted Linux distribution at a low price in direct competition with Windows based software.

As a countermove, Microsoft recently announced that it will invest a sum equalling €700 million over a three year period for assisting in the foundation of an IT university specialising in software development.

2.3.1 Japan

Also Japan has declared that it intends to study the options of usage of FOSS at a national governmental level. €440,000 has been set aside for an extensive study, due to be finished at the latest in March 2003. The Japanese government points out that countries such as the United States, Germany and China, already have on-going projects surrounding both the use and the evaluation of FOSS. But decisions cannot be made on the basis of what others have done, and it is for this reason that Japan wants to have a foundation for making a decision of its own concerning the usage of FOSS.

Microsoft is by far the leading supplier of software, and it has been suggested that other alternatives be looked at, mainly from a cost and security point of view. The study will not be used to make a choice of

operating systems, but instead present a neutral view of the pros and cons of the various alternatives. Thereafter, the national government and other authorities themselves will take a position on which alternatives to choose.

2.3.2 India

The government in India, one of the largest countries in the world, recommends use of FOSS. The Indian IT department has started work on introducing Linux in national government systems and in the school systems all over the country.

The objective of the department is that all national government agencies and institutions offer courses in the use Linux and development of software for the operating system. There are even plans for establishing Linux centres at universities. The government and business interests would finance the centres jointly.

The objective to support Linux and get it established in schools and universities is primarily financial. The operating system and complementary software are considerably less expensive than proprietary alternatives. The possibility to copy and distribute programs based on FOSS free of charge is hoped to serve as an impetus to an increase in use of computers in the country. It is also believed that there will be a wide global demand for development and support of Linux based applications. This is seen as a future possibility for export and revenue.

Microsoft totally dominates the market in India with over 90%. As in China, Microsoft has announced that they will invest large sums of money for educational purposes over a three-year period.

2.3.3 South Africa

South Africa has carried out a study where they have probed the possibilities of using FOSS instead of proprietary software. In certain respects, they have chosen the same policy as Great Britain. The study points out the following advantages of FOSS:

- The freedom to improve and change the software
- The ability to avoid dependencies on and lock-in effects of products that cannot be changed or altered
- The freedom of distribution
- Lower acquisition costs
- Less administration of licenses and control of software piracy

- Great advantages at a national level through a lesser dependence on foreign companies
- Less outflow of currency
- Stimulation of local business initiatives
- Individuals can have electronic access to information without demands for proprietary software and steep licensing costs

The study also states the following preconditions for introducing FOSS:

- Competence surrounding FOSS must be increased; among other factors necessary, there must be access to training and education
- No discrimination of either FOSS or proprietary software. Users must receive competent support in choosing the right product. Choice of and migration to FOSS must be based on sound business principles. No subjective appraisal should be allowed
- Demands concerning security and interoperability must be observed

The authors of the study are well aware of the problems involved in migrating to FOSS and they are not foreign to the idea of a national strategy or policy with recommendations favouring FOSS. Initiatives and pilot installations will be encouraged. "Implementation of carefully selected OSS in the public service should be seriously considered. The current environment demands, however, that this be accompanied by a well-planned change management strategy."

URL: <http://www.oss.gov.za>, <http://www.naci.org.za>

2.3.4 Peru

Congressman Edgar Villanueva Nunez is the man behind the proposition for a new law concerning the use of FOSS in the Peruvian administration. The proposition has become the object of much attention in other countries. The proposition stipulates "free programs or software shall be used exclusively by all State institutions in their computer systems and equipment." The term "Peruvian law" has become a source of worry for producers of proprietary software.

An important factor in Peru's ambition to have FOSS in the country's computers is a fear that proprietary software can contain back doors and other security risks. Furthermore, a law concerning FOSS can stimulate local software development and create conditions for a local IT business sector.

At a conference organised by the World Bank, Villanueva told of how he first came into contact with FOSS. It was when he was mayor in a small city

in Peru. 98% of the population did not have any access to a computer. He was faced with the problem of how to give the inhabitants this access and also modernise public administration, but it was difficult solving these problems at a local level.

Villanueva was elected to the Peruvian Congress and he then had the opportunity to act on these questions. He spoke with many software producers, even Microsoft.

"I proposed a law advocating use of FOSS in public administration," says Villanueva. Congress should have passed the proposition on June 18, 2002. On June 17, a letter to the President of the Peruvian Congress arrived from the American Embassy. The letter expressed concern over the fact that the Congress was about to pass a law on FOSS."

"Every parliament must have the right to discuss such questions without the interference of other countries," says Villaneuva. "It is every country's duty to promote its own development and growth. Free access to information for a country's citizens and national security questions are vital."

"I'm not an extremist, but there is a need for a law that governs and is founded on these principles. The proposition does not forbid proprietary software. But access to source code is important for public administration. Being able to have insight in how the system for vote tallying works, for example, is important to citizens.

The proposition has been accused of discriminating against proprietary software.

"That's not true," says Villanueva. "The State represents its citizens and it is tax revenue it's all about. This revenue should be used in an efficient way. The proposition does not negatively single out other software, but instead guarantees good market competition."

FOSS is, according to Villanueva, the only possibility for developing countries to have legal access to software products. 80% of all software in Peru is pirated. There is a lack of resources and the country has a considerable national debt. "It is difficult to understand Microsoft's position," says Villanueva. "Only 20% of their sales in Peru pertain to public administration."

Villanueva says: "We are not against Microsoft software nor proprietary software. It is only that it is not the best option available. Maybe it is so for the European countries, but not for us."

URL: <http://www.gnu.org.pe>, <http://200.37.159.7/paracas>

2.3.5 USA

An extensive debate is currently taking place in the United States concerning both use and consequences of FOSS. Many FOSS development projects are situated in the US and both the Free Software Foundation and the Open Source Initiative are American endeavours. Most licenses for FOSS are also written for the American market.

However, there is still a comparatively modest use of FOSS within the American public sector.

2.3.5.1 The defence sector - DoD

MITRE, a research institution in the US, has shown results from a study on FOSS within the defence sector in the United States. FOSS can be found in a considerable part of current systems, and has three important attributes, which make it suitable for protecting the national infrastructure, namely:

- The fact that there are carefully reviewed products such as OpenBSD means that the systems are less vulnerable to cyber attacks.
- The Department of Defense's own development work, resulting in some of the most advanced tools around for finding and analysing weaknesses in networks and systems, is based on FOSS.
- It is possible to quickly fend against new attacks on a system or network due to the fact that the software is open and easily modified.

According to MITRE, the most-used license, GPL, is extremely fit for this purpose of giving users the possibility to quickly modify software in order to counteract attacks from the outside.

MITRE also states that what is special about security in Linux is that the source code is accessible, that it has been the subject of external review, is quickly corrected in case of bugs, etc. According to MITRE, one cannot say that FOSS is more or less secure, but the fact that source code is open for review is a great advantage.

2.3.5.2 US Census Bureau

The US Census Bureau provides data and statistics on citizens and on the economy in the United States. The Bureau uses FOSS in its own

development work. Among others, the following applications are based on FOSS:

- State and County Quick Facts
- FedStats and MapStats
- National Economic and Financial Indicators for the International Monetary Fund

The Bureau has chosen FOSS because of its portability in a mixed environment, for not having to put up with delays in procurement situations, because it provides a consistent and dependable environment and for support and access to source code.

Systems development work proceeds quickly, maintenance costs are low and the applications are stable. One example of this is State and County Facts. There were no special budget resources for the development project. FOSS and existing hardware were used. Software such as Perl, MySQL, Apache and Linux were utilised. A proprietary solution would have cost almost \$41,000 in licensing costs for a production server alone. State and County Quick Facts has 23,000 visitors a day.

FedStats is a portal for statistics information and is available within the decentralised federal system for statistics. More than 70 different agencies and authorities collect the statistics used.

The statistics system at the US Census Bureau is the largest installation of MySQL in the world.

2.3.6 FOSS in developing countries

A major problem when using proprietary software is the prohibitive costs. One example from Vietnam shows that the cost for MS Windows plus MS Office are equivalent to more than a year's wages for a common worker. FOSS is therefore important when developing digital administration in these countries. Another argument for FOSS is that developing countries want to favour their own domestic software industry. Even the anti-piracy policies of proprietary software producers, contribute to support for FOSS.

At the current time, developing countries are strongly dependent on non-domestic software producers. There is a lack of specialists, knowledge about the area and support. To a certain extent this has prevented use of FOSS. For this reason, many individuals receive training free of charge in how to use these kinds of products.

The situation in the developing countries can be summed up in the following manner: There is a lack of resources. ICT is an important area, not

the most prioritised, but without a development in IT, the digital gulf between the developed and developing world will contribute to an increase in poverty. In order to get development started it is therefore necessary to use the free resources which can be found on the Internet. FOSS is subsequently an important strategic question for the development of these countries.

3 Present situation – public administration in Sweden

3.1 SMHI – Swedish Meteorological and Hydrological Institute

Within SMHI there is a wide range of IT solutions. An important part of the work done is centred on research in the fields of meteorology, hydrology and oceanography. Both within research and also for calculation of forecasts there is a need in many cases for a considerable capacity for calculations, and development takes place to a great extent in an international joint context. It is here that Linux and FOSS have become an important factor. Complex calculations are carried out in a Linux environment where price and performance are better than anything else. Before using Linux, this type of activity took place for the most part in different proprietary UNIX-environments.

In the development of the advanced production systems, which are available for the increasingly automated refining of information from observation and forecast data to customer-adapted products, FOSS has played an increasingly important role, for example in:

- Open Map for certain less complicated GIS-oriented applications
- Communication between systems via ORB (Object Request Broker)
- Java libraries
- Web servers
- A number of supportive tools for systems development, e.g. CVS for configuration management

The developers and researchers who work a great deal with Linux need an integrated environment, and it is for that reason that SMHI has chosen office software also based on Linux and OpenOffice. At the current time SMHI is migrating to StarOffice for these users. Experience up to this point shows that StarOffice is a better product than OpenOffice, but for obtaining support, SMHI is referred to Sun. The large community for FOSS, which can be reached via various websites and news groups, is not reachable to the same extent via Sun. Moreover, it should be noted that development takes place faster for OpenOffice.

Generally, experience from FOSS within SMHI can be summarised in the following points:

Positive

- The quality and stability of many applications are superior.

- Support via web pages and mailing lists/newsgroups create a knowledgeably large and efficient support organisation.
- Lock-in effects from vendors creating compatibility problems with previous product and product family releases are considerably less.
- Lower costs of procurement.

Negative

- Increased requirements for in-house competence.
- Development is often dependent on enthusiasts, and when they are not as actively interested, the rate of development for FOSS diminishes or stops altogether.
- Documentation is not as extensive.
- Weaker in certain areas such as advanced GIS applications and powerful database applications.

3.2 Swedish Premium Pension Authority (PPM)

PPM uses Linux for systems, which administrate pension funds – the account administrative system called Pluto. Earlier, HP-UX was used. The Linux environment has resulted in better performance and lower costs.

PPM has put work into finding a Linux core, which application suppliers can support. The core is being developed at a fast pace with new versions, and all suppliers must support the same version in order to be able to receive full support. RedHat is used, although not the latest version, in order to receive full support.

RedHat for Linux Advance Server offers a year's free support. RedHat has support centres in Holland. EMC and HP support their respective areas. PPM can therefore turn to its regular suppliers. This works to PPM's satisfaction.

As far as costs are concerned, the migration to Linux has been favourable. Furthermore, it is easier to retain competence by only having two types of servers. A great advantage is that PPM does not have to pay for support, since the supplier agreement with *Statskontoret* calls for 3 years of support with procured server products. The low cost also means that PPM can afford to have separate test and development environments that are identical with the production environment. This means that they don't have to have extensive support agreements, but can instead use their test environment as a back up if necessary.

PPM intends to continue with the migration and will change their Internet platform from HP-UX to Linux. At the present time they have 20-25 HP-UX servers.

Of other FOSS products on the market they have only tested StarOffice. Here they see potential problems, since the calendar function in MS Outlook is used extensively for booking rooms and facilities. For this reason, they will continue to use MS Office, but they realise that they can save money by a migration to an open alternative.

3.3 Swedish National Labour Market Administration (AMS)

AMS has carried out a report on the possibilities of replacing their current office suite with StarOffice. The reason for the study was that Microsoft's office products were too expensive plus the fact that there was concern about becoming too dependent on a single vendor. An exchange of ideas and experience has taken place with the Swedish National Police Board.

The work was concentrated in the following four areas:

- StarOffice in AMS' technical environment
- Adaptation of current templates and macros
- User experience
- Training and support

One important issue is the job of adapting current templates and macros. Templates in MS Word were deemed reasonably easy to manage, but AMS has a large number of Excel macros in use. The Word templates are managed centrally, but Excel macros could have been created in many parts of the organisation. The general agreement was that work with macros for spreadsheet applications would have to be done all over from the beginning.

Users formed a test group, which tried out the product for two weeks. The test group was mixed as far as tasks at hand and previous knowledge. The result of the test was that the more experienced users were more positive to StarOffice than the less experienced.

The technical evaluation showed that there were no problems introducing StarOffice in AMS technical environment. Since the product was open, there were a great number of advantages, e.g. integration with an LDAP-based catalogue, the possibility of tailoring menus for the needs of different users and database connections.

Training and support needs were not considered to be so extensive; AMS' own helpdesk/back office functions were regarded as being capable of this.

The experiences which AMS show are the following:

- Expensive to introduce as far as “good will” is concerned (i.e. end users must be convinced to change products).
- There are no technical obstacles to migrating to StarOffice.
- There is limited knowledge of the product on the market.
- AMS can do product adaptations.
- Difficult to communicate with others outside organisation because of differences in document file formats.
- Twofold environments necessary during a considerable period of time.
- Web based products are of cornerstone importance to AMS and this is not available in the current version of StarOffice.

AMS will wait to introduce StarOffice until a web-based version comes out.

3.4 Municipalities

Two students at the Blekinge Institute of Technology, Charlotte Kihlstrom and Mattias Kihlstrom, wrote a term paper in the spring of 2002 where a study was made of Swedish municipalities’ use of FOSS. A questionnaire was sent to all the municipalities with questions about how well they were acquainted with FOSS, to what extent it was used and also about their attitudes to FOSS.

The results showed that about 60% were well-acquainted with what FOSS is, 80% see it as something positive and about 30% use FOSS in some way. The answers also pointed out to the fact that better support, increased security and clearer information on what FOSS is are all important factors to deal with if the use of FOSS is to increase. Better security was what those who had a limited knowledge of the area mostly noted.

3.5 Malmo General Hospital (MAS)

Some of the clinics at Malmo General Hospital have started to use Linux as an alternative to Windows and OpenOffice/StarOffice as an alternative to MS Office. They use a solution built around thin clients based on Citrix. Experiences up to this point are positive. MAS’ ambition is for all computers to contain some alternative to more expensive licenses by the end of this year.

3.6 Linux in the schools

Skolverket, the National Agency for Education, organised a seminar in the spring of 2002 on Linux and FOSS in a school environment. As part of the

programme, experiences from the *Skolelinux* project in Norway and the *Gnuskoleprojektet* in Denmark were presented. A Swedish user group has continued with activities in the area. There are also discussions taking place, together with Swedish municipalities, on using StarOffice in the schools.

3.7 Current environment in public administration

The current environment in Swedish public administration is mainly based on Microsoft products.

On desktop or workplace computers there is almost a total dominance for different versions of MS Windows (Win98, NT4 and Windows 2000) along with MS Office (Office 97 and Office 2000) as office suite. MS Internet Explorer is almost the sole browser used.

There is a more heterogeneous server environment with a lot of UNIX-based systems, but there are also a number of systems based on MS Windows NT4 and Novell NetWare. The few Linux systems used are almost all used as web servers with Apache installed. Other FOSS-based software used is Sendmail and BIND.

Many administrations have plans either to upgrade their desktop computers to a newer version of Windows and MS Office – often due to the fact that support from Microsoft is not available anymore for older versions – or to evaluate some other alternative software, for example StarOffice.

3.7.1 Agreements

In most cases administrations and authorities have current agreements, which extend to the summer of 2004.

3.8 Projects and investigations within Swedish public administration

SHS, (*Spridnings- och hamtningsystem*) the Swedish national infrastructure for information exchange, is used for the exchange of information with public administrations. The European Commission has evaluated SHS as a tool for common projects. At the present, specifications for SHS are freely available. The question has arisen concerning the possibility of making SHS available in the form of open source code.

4 Why free and open software

4.1 "Why Open Source Software" – David A. Wheeler

One of the strongest arguments for using FOSS is to arrive at a greater independence as far as price and licensing conditions are concerned. In a situation with financial restraints, new and harder licensing terms and conditions and with software that comes more often than before in new versions, costs for a software environment are rising.

David A. Wheeler, an independent researcher and software developer, has written an uncommissioned and unique report containing information on the use of FOSS: "[Why Open Source Software / Free Software \(OSS/FS\) Look at the Numbers!](#)". The report contains an extensive number of URLs for on-going information and is continually revised.

The report covers areas such as:

- market share
- reliability
- performance
- scalability
- security
- TCO (Total Cost of Ownership)
- unnecessary fears
- usage reports

To sum up, Wheeler states that FOSS has a significant portion of the market within many areas. Worth noting is the fact that more than 65% of all of the servers on the Internet run Apache, often with Linux as an operating system. Wheeler also states that FOSS often has a higher degree of dependability and in many cases better performance in a direct comparison with proprietary alternatives. Scalability in the model for development of FOSS results in products being developed for a large number of computer platforms and environments. As far as security is concerned, an area difficult to measure and compare, the report states that security in FOSS is just as dependable and effective as in proprietary alternatives and that FOSS, when compared to proprietary software, is less affected by attacks and virus from the Internet. The cost structure is of a clear advantage, especially when focussing on direct costs involved.

Wheeler points out that in order to be able to achieve the potential advantages of FOSS, there will be a need to review one's procurement

process in order to be able to receive bids or tenders that also include alternatives which include FOSS.

One question, which shows up in different studies, according to Wheeler, is that of the risks involved in becoming dependent on a single vendor, who has control of a whole infrastructure. In Germany, for example, the use of FOSS is motivated by security reasons among other. The German Minister of the Interior has expressed the situation in the following way: "Monoculture is not good for security".

4.2 The advantages of FOSS products

In the FLOSS report, described in chapter 2.1.3, a comparison has been made of the use of FOSS in three countries, Germany, Sweden and Great Britain. Germany is the country with the most widespread use of FOSS, Sweden the lowest.

The advantages which users saw in FOSS were:

- higher stability
- high level of security
- none or low licensing fees
- possibility to modify source code
- ample access to IT specialists
- independence from major software vendors

One considerable advantage of FOSS is the ability to easily modify or change a product for a certain group of users, i.e. to make it simple and functional. One can create a specific development environment, an application for e-services for the general public or a customised desktop or workplace computer with modified functionality. Products become more operatively secure and easier to administrate and maintain. Initially, there are costs for development, but running costs are very low and total costs much lower than corresponding traditional proprietary alternatives.

But, unfortunately, the freedom to choose can scare off some prospective users. Many prefer packaged solutions. Therefore there should be place for systems integrators who can offer packaged solutions, based on FOSS, and which are supplemented by a well thought-through service for migrating from a proprietary to an open environment.

Although costs are high for the migrating from proprietary software to FOSS, most studies show that a migration, both in a short and long term, leads to considerably lower costs. Moreover, it is pointed out in the study that costs for support, service and maintenance are generally the same as for

proprietary software, but that these services can be carried out by local suppliers, leading to a positive effect on employment rates, tax revenues, etc.

It is important to take into consideration all related cost over a longer period of time when comparing FOSS and proprietary software.

4.3 Workshop on areas of use

A workshop was carried out within the working group for this report. The aim of the workshop was to identify, assess and prioritise what was influential and decisive when choosing software products and especially focussing on FOSS. What are the incentives for a changeover and why?

Note that the results from the workshop only reflect the view of the questions at hand by the individuals taking part.

The following questions were posed to the participants:

1. What factors are decisive for **choosing another** software product?
2. The selection of **areas of use** where FOSS is considered, can play an important role. Estimate what **effect** the use of FOSS can have on these areas.
3. Define which criteria are **influential/decisive** when choosing software.
4. Specify **advantages/disadvantages** of FOSS.

Answers to the first question resulted in a number of factors, which were then applied to the identified areas of use for FOSS.

The next task was to define what effect the use or implementation of FOSS has for the factors arrived at above.

The third task, criteria, were ordered for the identified areas of use and then compared to the advantages and disadvantages of FOSS in task number four.

The results showed that the working group saw a more or less positive effect from FOSS in all of the identified areas. The criteria for choosing a software product, which were generally considered most important, were security, stability, cost and that a product follows a certain standard. These criteria were also in agreement with those advantages, which are related to FOSS in other reports and studies.

4.3.1 Why change software?

The factors that were identified as most decisive for the establishment of a **change** in software environment were general factors that have no direct connection to FOSS.

- Freedom of choice – increased competition
- Long term development of product – technical lifetime
- Less dependency on single product vendor – avoid lock-in effects
- Product performance
- Product stability and dependability
- Correctly adapted functionality – i.e. the product's functionality satisfies the needs of the user
- Security
- Lowered costs
- Access to source code – for adaptation, modification and control over future development
- Access to support

It is interesting to note that these factors also include those, which were identified in the FLOSS report as positive factors of FOSS, where stability, security, cost, independence and access to source code had decisive importance for the use of FOSS.

4.3.2 Areas of use for evaluation

The following areas of use were deemed interesting, partly because they represent a very large share of the software which can be found in a government agency and partly because there either are in existence, or could easily be put together alternative software solutions in these areas based on FOSS.

- Server operating systems with
 - Linux and various BSD versions as representatives of an "open world"
 - Various versions and variants of Windows and UNIX as a proprietary alternative
- Desktop operating systems (PC) with
 - Linux
 - Microsoft Windows
- Office software including e-mail
 - OpenOffice/StarOffice with e.g. Evolution as e-mail client
 - MS Office with Outlook
- Web

- LAMP-based web servers, i.e. Linux based Apache web servers with MySQL database and scripting languages such as PHP, Perl or Python
- Microsoft IIS based solutions (Internet Information Server)
- Development environment
 - For example development environment Eclipse which is supported and used by Borland, HP, IBM, Rational, etc. (<http://www.eclipse.org>) or Mono which is an open implementation of the development environment for .NET (<http://www.go-mono.com>)
 - .NET – Microsoft’s new development environment
- Databases
 - MySQL and PostgreSQL
 - Oracle, Microsoft SQL Server, etc.
- Business specific systems
 - Customised systems for special purposes

One area which doesn't fit into the evaluation, but where there are interesting solutions based on FOSS is the area of security. For example, there are products and solutions for firewalls and VPNs (Virtual Private Network)

The following table shows the working group’s view of the effects which usage of alternative solutions based on FOSS could result in for a government agency.

Note that the table presents a view of how the working group analysed the current situation at the time the workshop took place (November 2002).

Areas	Server OS	Client OS	Office	Web	Devel. envir.	Database	Business specific systems
Correctly adapted functionality	+	+	+	+	=	+	+
Long term development and lifetime	+	+	=	+	=	+	+
Less dependency, no lock-in	+	+	++	++	+	+	++
Performance	=	+	=	=	=	=	+
Stability and dependability	+	++	=	++	=	=	+
Security	++	++	+	++	=	+	+

Lowered costs	++	+	++	++	+	++	+
Access to source code	++	+	+	+	+	=	+
Access to support	=	-	-	+	-	=	=
Freedom of choice	=	+	++	+	+	+	++

Table 1: Working group's evaluation of effects of the use of FOSS

- ++ Very positive effect
- + Positive effect
- = No effect
- Negative effect
- Very negative effect

The noted effect does not accordingly denote which product can be regarded to be the absolute best product in a given area of use, but instead the actual needs which a user in a public administration could be considered to have. A comparison could be made to the car industry where a Mercedes in all respects is considered to be a better car than for example a Volkswagen, but where the simpler (and cheaper) car perhaps corresponds to the actual needs of a car-buyer.

When directly comparing Microsoft Office and OpenOffice/StarOffice, the former stands out of course as a much more mature product with more functions and with a more well thought-through user interface. The same thing can be said for the area of databases, where Oracle, for example, has a product where the number of special items and functions exceed that which the open alternatives can offer.

Within the area of security, the estimated effects are based for the most part on the experience gleaned from proprietary products with a background of security disruptions, which are caused by defects in the software itself.

4.3.3 Criteria for selecting (open) software

The next part of the workshop was aimed at arriving at criteria, which are influential and decisive when choosing, or selecting software. Thereafter these criteria were ordered in relation to the above-named areas of use.

- Ease of use/user friendly
- Flexibility
- Functionality
- Procurement cost

- Supplier competence
- Platform compatibility
- Performance
- Scalability
- Standardisation
- Security
- Dependability and stability
- Access to drivers
- Access to services such as support, service, training, etc.
- Total cost / Life-cycle cost
- Openness / Access to source code

The result is presented in the following table. Note that they are ordered in a comparison with relatively equivalent software with about the same functions, and that the prioritising made of selection criteria is always situation-dependent.

	Server OS	Client OS	Office	Webb	Devel. envir.	Database	Business specific systems
Ease of use/user friendly		A	A				C
Flexibility	C			C	B	C	
Functionality	C	C	B	C	A	C	A
Procurement cost			B				
Supplier competence			B		A		C
Platform compatibility	C	B			C		
Performance	B			B	C	B	
Scalability				C		B	C
Standardisation	B	C	A	A	B	B	B
Security	A	B	C	A		A	
Dependability and stability	A	B		A	A	A	A
Access to drivers		C					
Access to services	B	A	C	B	C	C	B
Total cost	A	A	A		B	A	A
Openness			C	B			B

Table 2: Criteria for selecting software

Level A - The most important and decisive selection criteria

Level B - Important and desirable selection criteria

Level C – Selection criteria which are of some importance, but not directly conclusive

The last task in the workshop was to make a list of the positive and negative effects, which can be associated with FOSS. Note again that the effects reflect the participants' view of FOSS within the area of public administration.

Positive effects:

- simpler license management
- reduced dependence on a product, less risk for "lock-in" effects
- lower costs overall
- increased competition
- increased quality and stability
- increased activity on part of local/domestic businesses
- increased security
- open formats simplify communication with general public

Negative effects:

- possible need for extensive migration
- could lead to higher demands for in-house competence and maintenance within the agency or authority itself
- could be difficult finding the right product
- possible interoperability problems with proprietary software
- fewer available consultant and support services on the market at present time
- psychological resistance among decision makers

It can be of interest that the criteria, which the workshop viewed as being the most important for the selection of software –security, stability, cost and that a product follows a certain standard –, also are to be found as positive effects for the chosen areas of use.

4.4 Successful FOSS projects

The following are a few examples of successful FOSS projects. A more complete list can be found at <http://www.linux.org/apps/index.html> or at <http://www.sourceforge.com>.

4.4.1 Infrastructural applications

Apache is one of the world's most utilised web servers. Apache is often used in combination with Linux, MySQL and Perl/PHP/Python in the so-called LAMP concept. Apache is used in more than 65% of all web servers on the Internet today. <http://www.apache.org>

BIND is, without question, the most widely used name server (DNS) on the Internet. A name server's foremost task is to translate computer names (e.g.

www.statskontoret.se) to IP addresses and vice versa. Without DNS (and BIND!) the Internet would not work.

CVS (Concurrent Version System) is an often-used program for version control and program development. CVS allows several different developers to make changes at the same time in the same program code and then combine these changes and warn of possible conflicts. CVS is a client/server application and the central database is placed in a web server. Clients are available for almost all platforms: UNIX, Windows, Macintosh and even a Java based version. <http://www.cvshome.org>.

DHCP (Dynamic Host Configuration Protocol) is software for automatic configuration of nodes on an IP-based network. The most common use is in a LAN for assigning IP addresses. <http://www.isc.org/products/DHCP>

Open LDAP is an open implementation of LDAP, including applications for servers and clients along with development tools. <http://www.OpenLDAP.org>

Samba is a product which implements the SMB protocol (Server Message Block) on a UNIX based system. SMB is also used in e.g. LAN Manager and NetBIOS. Samba functions as a file and print server for both Windows and Linux clients. Samba can also emulate an NT 4 PDC server (Primary Domain Controller). <http://www.samba.org>

Sendmail is the most common MTA (Mail Transfer Agent) on the Internet. An MTA routes e-mail from one machine to another, often based on the SMTP protocol (Simple Mail Transfer Protocol). Sendmail includes functions such as mail routing, mail relay and anti-spam. It also supports protocols such as SMTP, POP and IMAP. Sendmail can also function as a client e-mail server and is also to be found in a proprietary version with added functionality. <http://www.sendmail.org>

4.4.2 Databases

MySQL is an SQL based database server with accompanying applications for administration, control and clients. MySQL is the most frequently used open source database on the Internet and is available both as open and as proprietary software with two different licenses. Support for almost all platforms including Linux and Microsoft Windows, FreeBSD, Sun Solaris, IBM's AIX, Mac OS X, HP-UX, Novell NetWare, etc. <http://www.mysql.com>

PostgreSQL is the other of the widely used SQL based database servers. PostgreSQL is often used for less complicated installations. There is not the

same range of support, documentation and services as for its "competitor" MySQL. <http://www.postgresql.org>

4.4.3 Desktop computers

Fetchmail is a widely used mail client with support for almost all mail protocols, all versions of POP, IMAP, SMTP and ESMTP. The application also supports IPv6 and IPsec along with a number of security functions, which handle link-state protocols such as SLIP and PPP for dial-up connectivity. <http://www.tuxedo.org/~esr/fetchmail>

GIMP (GNU Image Manipulation Program) is popular open source software for picture editing. It can be used for editing and touching up of pictures from digital cameras, but also as a general purpose drawing program and for converting graphics files between different formats. <http://www.gimp.org>

GNOME is a complete graphical user interface with window handling, file handling, control panel, etc. Moreover, there are a large number of applications adapted to GNOME. The project is very active and is supported by among others, HP, IBM and Sun. <http://www.gnome.org>

KDE is, like GNOME, a complete graphical user interface with window handling, file handling, control panel, etc. KDE also has a very large number of adapted applications. <http://www.kde.org>

Linux as an operating system for desktop/workstation computers is not yet as common as for servers. But much work has been put into improving the user interface and both GNOME and KDE have been improved during the latest six months. It is in the first place Mandrake, SuSE and RedHat among the commercial distributors and Debian, who are the most popular. <http://www.desktoplinux.com>

Mozilla is a web browser, which has some program code identical to that within Netscape Communicator 6. After some years of problems, the Mozilla project has improved its program code and can now offer a complete suite of programs with support for both Linux and Windows. <http://www.mozilla.org>

OpenOffice.org is a complete office suite for word processing, presentations and spreadsheet. Representatives from Sun administer the project and Sun contributes to a large part of the development. The project has developed their own XML-based file format and also support for MS Office file formats, although not 100%. Works on both Linux and Windows platforms. <http://www.openoffice.org>

StarOffice from Sun is built on the open source software OpenOffice.org with additions of proprietary software for hyphenating, database connectivity and an increased number of filters for import and export of documents along with support services from Sun. StarOffice uses the same XML-based file format as OpenOffice.org. <http://www.sun.com/staroffice>

WINE (Wine Is Not an Emulator) is an open implementation of the Windows 3.x and Win32 APIs. It is a 9-year-old project, which started with the aim of building an alternative Windows compatible implementation consisting of 100% Microsoft-Free code. WINE makes it possible to run most Windows programs in an Intel-based Unix/Linux environment, without any need of modifications.

Ximian Evolution is a mail client, much like MS Outlook. Evolution includes functions for e/mail, calendar, booking of meetings, contact information, etc. By using a proprietary add-on, the mail client can function directly with MS Exchange 2000. Support is also built in for IMAP4, POP3 and SMTP. http://www.ximian.com/products/ximian_evolution

4.4.4 Security

Astaro Security Linux is a relatively new firewall solution, offering among other things stateful packet inspection filtering, content filtering, user authentication, virus scanning, VPN with IPSec and PPTP. Included is also a web based management tool. The firewall is based on a "hardened" version of Linux 2.4.x. <http://www.astaro.com/>.

FreeS/WAN is a popular open source application for building VPNs (Virtual Private Networks). It uses IPSec and common encryption algorithm such as 3DES and AES along with the option to compress data in traffic. The latest version has a new function called Opportunistic Encryption, which uses functions in secure DNS for the setting-up and configuration of secure VPN links. <http://www.freeswan.org>

OpenSSH is an open version of the last free version of Tatu Ylonen's SSH, with all patent-protected algorithms removed, all known security bugs fixed and supplemented with new functions. OpenSSH is therefore communication software for secure, encrypted connectivity. It offers a secure alternative to software such as Telnet, rlogin, FTP, etc. Moreover, OpenSSH offers VPN-functionality and a number of authentication methods. <http://www.openssh.com>

4.4.5 Servers and administration

FreeBSD is an open UNIX-like operating system developed from BSD-Unix. The foremost aim of its developers is to be able to deliver a stable and high-performance a system as possible and to place a minimum of demands on hardware. <http://www.freebsd.com>

Linux as server operating system is as accepted today as UNIX or MS Windows, and companies such as Dell, HP and IBM position Linux as a full alternative to their own UNIX systems and Windows with the same level of service and support. The most common Linux distribution is Redhat, closely followed by Mandrake, Debian, SuSE and Slackware. <http://www.linux.org>

OpenBSD is also an open UNIX-like operating system based on BSD-Unix. The foremost aim of its developers is to produce the best operating system for security applications, <http://www.openbsd.org>

Webmin is a web based management tool for system administration of a large number of server applications such as DNS, Apache, Sendmail, Samba, etc. Webmin, written for the most part in Perl, consists of a simple web server and a number of CGI programs for configuration, administration and monitoring. Works on – more or less – all UNIX-like operating systems. <http://www.webmin.com/webmin>

4.4.6 Development

Perl (Practical Extraction and Report Language) is a high level programming language, developed in C and a number of less complicated scripting languages. It is well adaptable for system applications for administration and management, database applications, networking programs and web development. Development is easy and quick with Perl, and as a complementary source there is "The Comprehensive Perl Archive Network" (CPAN) with access to hundreds of Perl modules. <http://www.perl.org>

PHP is a very common scripting language, often used in the development of web applications. There is also extensive support for databases, XML, LDAP, IMAP and Java; its syntax has similarities with C, Java and Perl. PHP works in most environments, <http://www.php.net>

Python (The name comes from Monty Python's Flying Circus) is the third very common scripting language. It also comes in a fully Java-integrated version, Jpython. Python, like Perl and PHP, has a large number of system calls and class libraries. <http://www.python.org>

4.5 Security aspects

As far as data security is concerned, FOSS is considered to be less vulnerable than proprietary software simply because the source code is open and accessible. And in the case of proprietary software, trust must be placed in the manufacturer that it has constructed a secure application. Neither known or unknown backdoors nor other security related deficiencies should be retained in a version that has been delivered to a user. For example, it has been established in a number of reports that it is much more usual for a website based on proprietary software to be affected by security related mishaps. It is customary when developing FOSS that all developers are encouraged to carry out security related tests of the product. Since the source code is exposed, there is, in principle, an unlimited group of reviewers and all bugs that are discovered are revealed openly.

What kind of software development that can offer the best preconditions for a secure product is something, which has been discussed, in countless situations and contexts. Often the terms "security by obscurity" and security by openness" are used for the two alternatives. But up to this point, the general view among security professionals has been that to arrive at optimal security, the open model is the best. This is especially evident in the use of encryption algorithms, where the algorithm itself, in order to be trusted, should be open for review.

It is also the high demands placed on security and openness that have made Linux, and in the first place, Free/OpenBSD, the operating system that has been chosen by a considerable number of agencies and governments all over the world.

Objections to FOSS often include criticism that there are no guarantees that the product is dependable. But that guarantee also, in general, applies to proprietary products. Indirect damages such as unrealised profits or costs for restoring lost data are in principle, never compensated. And for the direct damages, which a company, according to law, cannot disclaim, a limit is set to only a small sum. Most user licenses (EULA – End User License Agreement) for proprietary software exclude also explicitly all responsibility for deficiencies in the software itself. The following is an example from Adobe Systems:

“DISCLAIMER OF WARRANTIES: YOU AGREE THAT ADOBE HAS MADE NO EXPRESS WARRANTIES TO YOU REGARDING THE SOFTWARE AND THAT THE SOFTWARE IS BEING PROVIDED TO YOU "AS IS" WITHOUT WARRANTY OF ANY KIND. ADOBE DISCLAIMS ALL WARRANTIES WITH REGARD TO THE SOFTWARE, EXPRESS OR IMPLIED, INCLUDING, WITHOUT

LIMITATION, ANY IMPLIED WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, MERCHANTABLE QUALITY, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.”

“LIMIT OF LIABILITY: IN NO EVENT WILL ADOBE BE LIABLE TO YOU FOR ANY LOSS OF USE, INTERRUPTION OF BUSINESS, OR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND (INCLUDING LOST PROFITS) REGARDLESS OF THE FORM OF ACTION WHETHER IN CONTRACT, TORT (INCLUDING NEGLIGENCE), STRICT PRODUCT LIABILITY OR OTHERWISE, EVEN IF ADOBE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.”

4.6 The public perspective

Together with the demands from government and authorities concerning a high degree of security lies also the obligation of a public administration to supply information to a general public. The general public has both a democratic right to information and a right to insight in the public agency or administration. This right includes, in part, the right to knowledge on how data and information is used in the administration’s computer systems, including what software is used. Examples from areas such as vote tallying and tax calculation make this right evident.

The FLOSS report states four main reasons for why public organisations and authorities consider either making policies or initiating direct implementations of FOSS: independency, costs, security and openness.

The report also points to the fact that the information which authorities make use of is owned by the public and by the citizens themselves, and that it is the responsibility of the authorities to:

- Guarantee free access to public information.
- Maintain preservation of public information.
- Guarantee a secure management of public information and that, which is obtained from the public itself.
- Avoid unnecessary expenses.

It is therefore very essential to use open standards (see next section) and to avoid lock-in effects (see section 6, The lock-in effects of vendor dependence). The possibility for the public for electronic access to information must not be directed or controlled by individual products or vendors.

5 Open formats and standards

Free and open source software preferably makes use of open standards, but there is nothing explicitly said about this in the various types of licenses for the software. The licenses regulate, in principle, only the relationship between the originator and the user and do not deal with what type of standard is used or is to be found in the software.

The standards that are of interest in this context are definitions of file formats and standards for the exchange of information, i.e. protocols and formats for communication between different applications and systems.

The term standard means an accepted norm for a certain activity, event or occurrence, or a variation thereof, which is commonly used or accepted. There is, on the other hand, no clear definition of the term open standard, but generally a standard is considered open if:

- Anyone can use it or take part in work on creating the standard.
- It is arrived at in generally accepted and open environment.
- Development takes place according to a model of consensus, and not led by one single player.
- The specification is published freely and is available without cost or only at prime production cost.
- Documentation is published freely and is available without cost or only at prime production cost.

There should also be some guarantee that the standard will be supported and maintained.

Examples of open standards are HTML and TCP/IP. HTML is published by World Wide Web Consortium (W3C), an independent body of companies and organisations and TCP/IP is published by Internet Engineering Task Force (IETF). These organisations also accept the responsibility for the further development of their standards. Both W3C and IETF fulfil the requirements listed above.

It is logical that FOSS uses open standards. In the first case it fits the basic values of FOSS and also it entails that a publication of FOSS is also an open publishing of contents of the software.

It is important to check to what degree a product is really based on open standards. If the product does not follow the open standards one is at a risk of having in time to change over to another solution, with increased costs as a result.

5.1 Open file formats for office software

The problem with proprietary office programs is that when they create a document file, they use a special representation for format, structure and content that are built-in the software itself. This results in a document file, which can only be recreated 100% accurately by software which recognised the original representation.

It is these proprietary document formats, which tend to be an obstacle for companies and government agencies. Many different applications need to be able to use what is contained in the document files, and it results in serious problems if these are not compatible with the document formats of other applications. Another aspect concerns the legal requirements for archiving, where information should be kept for as long as possible, preferably for "a limitless time". This, in principle, demanded that an open, standardised document format be used in order to avoid readability problems in the future.

There is, therefore, a great need of a standardised and open document format for storage, document exchange, archiving, etc., rendering possible an exchange of information between applications and systems.

Even if the standardisation process is open and produces open file formats, the work itself and the standardisation body must be accepted by all if the standard is to be used and have impact. The role, which XML has acquired within the area of document, formats supports the persuasion that an accepted and acknowledged group such as W3C (World Wide Web Consortium) or OASIS (Organization for the Advancement of Structured Information Standards), should have the responsibility for producing such a standard.

Furthermore, the working group, which works with the standardisation of document formats, ought to include representatives from all the major players and interested parties. A working group with participants from among others Microsoft, Sun, Corel and representatives from open software projects such as OpenOffice.org would ensure a favourable result, which all parties could support.

Within OpenOffice.org there is at present work being carried out for creating an open XML-based file format for office documents and an open reference for the format. It is the same format, which also is used in Sun's office product StarOffice.

The project has stipulated a number of firm demands:

- Full functionality and function as a format for exchange.

- Structured content, which can use the structuring properties of XML such as elements and attributes.
- The file format must be fully documented and not contain any hidden functions.
- OpenOffice is to be the reference implementation for this file format.

Furthermore, the following goals have been defined:

- The file formats should be developed in such a way as to be acceptable by all and to be managed and developed as an open source software project.
- The file formats should be well matched with all types of office documents (documents, spreadsheets, presentations, etc.)
- The file formats should as far as possible reuse portions of each other in the various office programs.

In order to ensure that work within OpenOffice.org will result in a generally accepted file format for office applications, Sun has recently (November 2002) decided to release the specifications for the XML-based file format to OASIS. A working group has been formed (The OASIS Open Office XML-Format Technical Committee) with representatives from among others Sun, Corel, Sony, Boeing, Drake Certivo and Abortext. Microsoft, however, is not taking part in the working group, although it is a member of OASIS.

Microsoft, which in the forthcoming release of Office 11, will offer an XML-based document format for Word and Excel, will put off its participation in the group. Microsoft also says that it will support all file formats which are based on XSD 1.0 (XML Schema Definition is a W3C standard).

5.2 Interoperability with other products

Inadequate interoperability between applications and systems are one of the main reasons for why businesses and organisations hesitate to change over to free and open source applications in the computers in their working environment.

Today, Microsoft, with a market share within certain product areas of around 85-90%, enjoys a distinctly dominating position and the document formats in MS Office have become something of a de facto standard.

A dependency of this kind means also that upgrades to new versions are not prompted by actual needs, but rather by the product cycle of the supplier. In many cases a software upgrade results in having to buy new hardware, leading to higher costs.

All attempts, on the other hand, at breaking out of this vicious circle entails a substantial amount of work, resulting in many customers accepting the situation as it is. The longer this kind of situation remains, the more difficult it is to break away. The end result is a situation where the vendor can dictate his own conditions.

However, it seems as if more and more users have taken notice of this situation and voiced a clear protest with a focus on increased costs, less favourable purchasing agreement conditions and increased lock-in. The result is that large numbers of users have started to look for an open file format for office applications and have become interested in alternative products based on FOSS.

6 The lock-in effects of vendor dependence

By "lock-in" we mean in this context, mechanisms used to keep a customer tied to a product or a vendor.

The IT sector is no different than most other business sectors: suppliers and vendors attempt to arrive at a deliberate degree of lock-in with the purpose of keeping a customer and securing his continued business. As a customer, it is impossible not to be affected in some way or the other, but increased awareness also means increased possibilities of containing and limiting possible damage.

Lock-in is a serious problem in situations concerning the procurement, operations, and maintenance of complex IT based systems. The problems are caused by costs for upgrading and system changes, the choices made in one's surrounding environment along with how one deals with previously made decisions.

A careful review of the costs for lock-in together with relevant requirement specifications when procuring new systems – before the effects of locking in can take place – constitutes an efficient way of lessening the negative effects of locking in. Furthermore, open standards offer a possibility of minimising vendor or product dependence, given that the implementation of the standard does not include one's own additions. Free and open source software can decrease the risks of locking in by using open standards and interfaces.

This chapter is about the various mechanisms, which exaggerate lock-in and how the risks and the consequences of locking in can be dealt with.

6.1 The mechanisms of lock-in

Lock-in is related to *changeover costs*, limitations caused by *previous decisions* and so called *network effects*.

6.1.1 Changeover costs

With the exception of staples, a change from one product or vendor to another always means higher costs, either for the customer or for the new supplier or vendor. Since the additional costs a customer is prepared to pay for a product normally are limited to what it costs to change over to an

equivalent product, many vendors are skilful at creating a situation where the costs for changing are considerable.

Changeover costs can be sorted into seven different categories.² It is important to know about and to be able to identify these so that lock-in doesn't come as a surprise.

6.1.1.1 Procurement of products with a long actual lifespan

When changing from one product to another there are additional costs for replacing earlier investments in products with a long actual lifespan. These costs decrease in tune with the value of the original product, but by making available service agreements, upgrades, add-on products, etc., suppliers often attempt to prolong the lifespan of a product. The same tactics are often used to generate costs even for products with a short actual lifespan.

6.1.1.2 Replace knowledge

Costs for learning to use a new product along with the decline in profits during the learning period is also a part of the costs for changeover. These costs increase over time, since knowledge of and familiarity with the old system increase.

6.1.1.3 Conversion of information and data

Because products often are incompatible with each other in a changeover situation, there are additional costs for converting files, databases, and other information artefacts from one product to another. The longer a system is in use, the more information and data will be tied up in the system. This type of changeover cost increases, therefore over time.

Even if there is a possibility of automatic conversion, there are costs involved, caused by error risks, configuration and preparatory work and also time itself, since the system must be "down" during the conversion process itself.

6.1.1.4 Specialised vendors

Dependency on an individual vendor who supplies highly specialised products can, over time, create a lock-in effect if competing vendors go out of business or lose their ability to offer compatible products.

6.1.1.5 Costs of finding an alternative

² C. Shapiro and H. Varian, *Information Rules: A Strategic Guide to the Network Economy*, Harvard Business School Press, 1998

The cost of finding alternative suppliers (whether they are specialised or not) new procurement processes, signing agreements, risks that a new supplier does not fulfil his undertakings, etc. must also be considered lock-in effects.

6.1.1.6 Customer loyalty programmes

Corresponding programmes, the equivalent of airline "frequent flyer" plans, along with other benefits, especially those that accumulate over time, result in costs for changing from one vendor to another and are used frequently by vendors in order to create lock-in effects.

6.1.1.7 Agreements

Finally, agreements often bind a customer to a certain vendor and vice versa. Changeover costs in the form of damages to be paid if a buyer breaks his agreement with the vendor are a by-product of the mutual insurance that an agreement entails.

6.1.2 Network effects

The term network effects means those effects, which contribute to increasing the value of a product, the more it is used. The classic example is that of the telephone that only became the valuable implement it is today, because the number of users increased greatly. It should be pointed out that network effects also affect systems that are not tangible; language is an example of a non-tangible network.

The lock-in effect for products whose value is based on a network effect have an intrinsic problem of co-ordination – it is not meaningful for a customer *on his own* to change to a better product, since the network effects support the most popular solution.

6.1.3 Previous decisions

The lock-in here is caused by past decisions, made on the basis of insufficient information, but which have been needlessly set in cement because of prestige or other irrational mechanisms. In microeconomic theory, this phenomenon is called *third-degree path dependencies*.

6.2 How to deal with lock-in

Lock-in effects are almost impossible to completely avoid, except for products, which have become staples - completely comparable, exchangeable and with a large number of competing suppliers battling against each other in hard competition.

The situation can seem gloomy, but by being well aware of the different costs which are caused by lock-in, it is possible to estimate the price of the lock-in effects, thereby making it easier to assess different vendors' "compensation" for lock-in and to decide which countermeasures are reasonable to carry out.

6.2.1 Well-achieved procurement

Since lock-in effects can weaken the customer's bargaining position in times of change, it is important to place relevant demands on open standards and interfaces when making a requirements specification. A well-achieved procurement is therefore a successful means of minimising the detrimental effects of lock-in. Shapiro and Varian provide the following practical advice for a procurement situation:

- Be aware of subtle lock-in. Even lock-in - which at the beginning can seem harmless - can grow to be a serious problem when in time one invests in add-on products and increasingly large amounts of data stored in proprietary formats and databases.
- Take it easy. Study the alternatives at hand and the consequences before making a decision on an individually selected choice of product or supplier.
- Always leave your door open for alternative possibilities. Prepare your procurement in an orderly manner.

6.2.2 Systems architecture

In situations when one can influence the architecture in an IT-system oneself - that is to say in circumstances where one develops in-house or contracts for development - one can, by preparing for interchangeability, minimise costs for change and therewith the degree of lock-in.

The higher costs for this kind of preparation can sometimes be hard to motivate if a changeover at the present time does not seem to be probable or desirable, but it is important to remember that lock-in gives room for the vendor to raise prices for service agreements, add-on products and other

things in the future. Future negotiations can be made easier for the buyer if wise decisions concerning systems architecture are made at an early stage.

However, be careful not to exaggerate work on a "perfect" systems architecture, since that investment – if it leads to less than optimum results – can constitute a damaging lock-in, in and by itself (see above: 6.1.3, Previous decisions).

6.2.3 Free and open software

Free and open software is often presented as being a solution to the problem of lock-in. The right to access, to modify and freely distribute program code for software systems does not eliminate automatically the effects of lock-in. The cost of changing from one system to another is still there and it is far from completely true – besides in certain fields – that the right to look at, change and redistribute program code increases one's number of options.

The attention, which FOSS has received lately, when even commercial companies invest extensive resources in the area, can be seen from a perspective where the major players are fighting for the right to benefit from lock-in effects. In a market where the software itself lacks direct possibilities for commercialisation, new opportunities arise in the form of adaptive work, integration services and maintenance.

Whether or not FOSS is a remedy for lock-in must, seen in this light, be judged separately in each case.

6.2.4 Open standards

Open standards have, without a doubt, been a strong contributing factor to a minimising of direct lock-in effects and a high degree of competition within many areas. GSM is a well-chosen example of how consumers as well as suppliers and vendors in the area of mobile telephony have been profited from standardisation. The opportunities for standardisation differ, however, between different areas of technology, due to prevailing market strength of a player and competitive vendor/customer situations.

When choosing the proposals and bids of various vendors it is of course suitable that special attention be given to assessing how applicable standards are met. Here one should be especially attentive to the risk of proprietary additions to open standards. A vendor can often say that it supports a standard completely, but append a number of enhancements which lead to lock-in effects if they are used. The value of these enhancements must be weighed against the costs of lock-in. FOSS minimises, of course, the

possibilities for a vendor to distort standards in this way, since all other vendors automatically have access to the enhancements.

7 Using free and open source software

Several different studies show that Linux is becoming more and more common as an operating system within companies and government agencies all over the world. In the first place, Linux replaces proprietary Unix and Windows platforms on servers. Studies show that Linux is the fastest growing operating system on the market today.

Major systems suppliers and integrators, such as Hewlett-Packard and IBM, put Linux on an even par with earlier traditional operating systems and offers support and consultancy services at the same level. IBM goes as far as choosing Linux as a Unix based operating system for certain system solutions instead of its own AIX. Most often a sort of partnership is established with one of the leading Linux distributors such as RedHat or MandrakeSoft and tests are carried out to ensure that applications and hardware are compatible and work together without any glitches. Both IBM and HP dedicate considerable resources in the development of Linux.

Even other application providers develop Linux based versions of their own software. Oracle is one such example. At Oracle it is considered just as important to make available software versions for Linux as well as any other operating system.

In some areas, Linux dominates the market for Internet servers: Web servers and large server clusters with stringent requirements for calculation capabilities.

The latest development of the user interface with windowing systems like GNOME 2.x and KDE 3.x along with office programs such as OpenOffice has made Linux a strongly viable alternative even for desktop computers. Sun has stated that they will also use GNOME for Solaris and will contribute to the development of both OpenOffice and GNOME.

7.1 The market for FOSS in Sweden

One common motive for not using free and open source software on a broad front is the fact that there is not the same range of vendors supplying services such as software and operational support. Even if the number of such vendors is steadily increasing, the scope is still limited when compared to the leading proprietary software vendors. However, the fact that suppliers such as IBM, Sun, HP, Dell, etc. have clearly voiced their full support for Linux and other free and open source software products, has made a noticeable impression on the market and created an increase in confidence in open alternatives.

In order to arrive at a more detailed and comprehensive view of what is offered on a national perspective, *Statskontoret* carried out a market survey in December, 2002 with the intention of publishing the results in a separate supplement to this study and also to present the same material on *Statskontoret*'s website – www.statskontoret.se.

7.2 Finding the right product

Finding what you're looking for in the enormous selection of free and open source software that can be found in portals such as Sourceforge and Freshmeat, can, at first, seem alarming. If you are looking for products other than the well-known software, then the following tips might be of some help:

- Check that the project and the software are well documented.
- Check the version of the software, which should not be less than 1.0. The longer a product has been active in a project, the less risk there is of running into problems.
- Study the on-going activities in the project, e.g. via mailinglists or number of project members.
- Even if the source code is always accessible, it can be simpler (and more secure) if the software is pre-packaged (and compiled) and tested for the chosen operating system.
- Test – if possible – the system first in an isolated environment

The above-mentioned criteria for selection can facilitate an introduction of FOSS. It should be possible to use the criteria for making a cost analysis for total costs over a fixed period of time.

7.3 Can FOSS and proprietary software be combined?

It is firstly the uncertainty of how FOSS, licensed with GPL (General Public License) can be combined with proprietary software, that can cause problems.

Most vendors of commercial software are completely opposed to using GPL licensed software, because of the fear of becoming "smitten" by the licensing conditions that stipulate that even derivative software must be published as FOSS.

In chapter 8.3.2, the differences between different licenses are listed in a table where it is also noted that GPL forbids reuse of source code in software that is distributed using any other kind of license than GPL. But there are not restrictions for the use of GPL-licensed software together with e.g. proprietary software, as long as they are independent of each other.

GPL-licensed software can be run on a computer with a proprietary operating system. Conversely, proprietary software can be run on a computer with a GPL-licensed operating system, for example GNU/Linux.

GPL-licensed software can also be linked to proprietary applications, as long as the software do not affect each other's design or structure. One example is the use of UNIX pipes, which allow the transfer of data between system components.

Furthermore, the licensing conditions for GPL are only applicable for distribution of derivative software. As long as one modifies the software for one's own private use, the conditions do not apply.

7.4 Migrating desktop computers

FOSS installed on desktop computers will probably not have any real impact until interoperability with MS Office is adequate. Using Linux as an operating system doesn't automatically mean that one has to stop using Microsoft's office software, since there is the possibility of creating a Windows compatible environment with the help of applications such as WINE/CrossOver (see section 4.4, Successful FOSS projects). In a migration phase, this can be a valuable alternative and can even be used for individual users who are dependent on a certain functionality in MS Office, but license requirements are the same, so there is nothing to gain in a financial perspective, and lock-in problems with a proprietary file format also remain.

Migrating to OpenOffice/StarOffice does not mean having to change operating systems, since there is support for both Windows and Linux (plus many other operating systems). This can make it easier for those who plan on moving to a Linux based computer some time in the future. Open Office/StarOffice has support for import and export of MS Office file formats, but the interoperability between documents and file formats is not yet fully complete, and never will be until Microsoft chooses to publicize the document and file formats for MS Office. It is therefore interesting to follow the work that OASIS has initiated in producing an XML based open file format for office applications with OpenOffice/StarOffice file format as a basis. Using OpenOffice/StarOffice involves substantially lower licensing costs and most reports indicate considerable savings, even if the migration itself can involve a large amount of work.

In order to determine if a migration to a Linux based desktop is the correct strategy, one should first consider the following points:

- How extensive is the degree of dissatisfaction with licensing and contract conditions from current vendors and what importance does one place on the fact that a proprietary solution can entail lock-in effects and dependencies?
- How large amount of "old" documents have to be re-used and do document templates have to be migrated?
- How easy is it for normal users to learn to use and understand a new PC environment and what competence is needed in the IT department?
- What costs are involved if one chooses not to migrate?
- What are the lifecycle costs and how often must software and desktop computers be upgraded and exchanged for newer?
- Can the conditions of one's current contract be changed in the light of the "threat" from alternative free and open source software?

A large organisation or government agency often has a considerable number of applications, which must be run on a desktop computer. Many of these (in certain cases, most) are Windows based and many are not available for Linux. These applications either have to be replaced with another alternative or, if possible, adapted for Linux, perhaps with help from software such as WINE/CrossOver. Even if many applications are web based, it is unfortunately all too common that they only work with specific browsers.

When evaluating alternatives with Linux based PCs and FOSS, one should choose one of the more popular Linux distributions such as RedHat, Mandrake, SuSE, OpenOffice/StarOffice for office software, Mozilla as a web browser and for example Ximian Evolution as an e-mail client and for connectivity to MS Exchange. Furthermore, one should consider the need to use or the possibility of using products such as WINE/CrossOver in order to create a Windows compatible environment in Linux.

There are also several interesting Linux based solutions for thin clients for Linux, UNIX and Windows based server applications.

In addition to free and open source software, there is a large and steadily increasing number of proprietary applications made for Linux.

7.5 Need for services

As previously mentioned, the variety of FOSS must be supplemented by a qualitative range of services in order to have a greater market impact. Examples of such needs:

- Prestudy
- Analysis of current situation
- Needs analysis
- Requirements specification
- Purchasing/procurement support
- Project management
- Implementation
- Development
- Etc.

Implementation and integration are two important areas in connection with the purchasing and introduction or launching of software. When installing and configuring, the required and promised functions must be assured. The new software must function in the current technical environment and sometimes there is a need for adaptation. It can be a great advantage to have access to source code and to have the right to make one's own modifications in the software.

The need for **training**, e.g. in the use of certain software and courses of the IT department in the management and administration of the software, does not differ from proprietary software. At the present time there is a fairly extensive range of courses covering popular FOSS such as Linux, Apache, etc.

Support and service for FOSS is offered mainly via the respective software project's community on the project's website, but can also be obtained from specialised companies and software distributors in the form of commercial services. Also here, the range of support and service is mainly geared to the most popular FOSS products.

8 Financial and legal aspects

This chapter describes business models for FOSS, how the public sector can benefit financially by using FOSS and also market competition related questions and suggestions for measures that can be taken.

8.1 Business models for FOSS

An interesting question is whether or not it is possible to make a profit on free and open source software. Since access to the workings of the software itself, the source code, is free of charge, any business model must be geared towards value-added services and products.

The development of FOSS is financed in part by interested parties, where ideological, political and commercial players are to be found. One important contributor is the public sector, which, by means of funds for research and development can contribute to the development and adaptation of FOSS. This occurs for example in the US, in the European Union and especially among developing countries.

Other interested parties who contribute in the form of resources, often as work time contributions, are commercial companies, which sell products, or services that are in some way connected to FOSS. For example, large corporations such as IBM and HP contribute to the development of Linux, and Sun to the development of OpenOffice.

Successful business models based on free and open source software emanate from one or more of the following areas:

- Software distributions – the sale of a packaged product based on free and open source software.
- Development and sales of in-house developed product.
- Added-value sales – free and open source software is used in order to support the sale of one's own supplementary products, such as other applications and hardware.
- Services – support, training, consulting, etc.
- Accessories – literature, etc.

8.1.1 Software distributions

The business model is based on the sale and packaging of FOSS. The basic presupposition is that users are willing to pay a certain (small) amount for access to the software. The distributor charges for the packaging itself,

which often includes CDs and manuals, but does not charge for the FOSS itself. Sometimes the product is also packaged together with proprietary software in order to give a user more value of his money.

This is the most common business model for FOSS. There are a large number of commercial distributors of Linux. The most popular are:

- RedHat (USA) – www.redhat.com
- MandrakeSoft (France) - www.mandrakelinux.com/en/
- SuSE (Germany) – www.suse.com
- SCO (prev. Caldera, USA) – www.sco.com
- Conectiva (Brazil) – www.conectiva.com
- Turbolinux (Japan) – www.turbolinux.com

The last four in the list, SuSE, SCO, Conectiva and Turbolinux, have formed a cooperative scheme for joint distribution. The scheme follows Linux Standard Base, known as UnitedLinux – www.unitedlinux.com.

Free Standard Group has an objective of creating a common base for Linux. The result will be a Linux Standard Base (LSB). One important driving force is to avoid the divergence that happened in the development of UNIX. By standardizing Linux, development is made considerably easier for software developers. LSB includes among other things, documented specifications and series of tests. Certification is carried out by an independent party.

RedHat, the largest Linux distributor, has 22 offices around the world with a total of 600 employees. Their business model is based on a software packaging, which also includes support, technical assistance, systems administration and training. Cooperation takes place with market leading partners such as IBM, HP, etc.

Examples of other FOSS, which is sold, prepackaged:

StarOffice from Sun, a prepackaged version of the FOSS-based office application OpenOffice.org, including proprietary software for spell checking and database connectivity, added fonts, filters, templates and support services.

ThinLinc from Cendio, a terminal server for thin clients based on FOSS such as Open SSH, TightVNC, Apache and Webmin together with Cendio's own, in-house developed software.

8.1.2 Development and sales of in-house products

One kind of sales of FOSS is when a company, which sells a software product, also has the main responsibility for its development and consequently becomes the company, which has the best competence concerning the product. It is a great advantage when selling added value services such as support, help to end users, training, special adaptations, further development, etc., since they are the most knowledgeable concerning the source code. Having the best knowledge is also a precondition for success since others can also use the open source code and develop it further or offer competing services. As long as a company is market leading in this respect, its business model remains viable and the company can make money on its own competence.

One special case is when a company also has the ownership rights to the open source software in question. This means that ownership of all development contributions (improvements in the source code), which are submitted, to the company, also go to the company. As an alternative to FOSS, a company can opt to distribute it with a proprietary license with a fee involved for its use.

MySQL is a good example of a successful Swedish company whose business idea is based on development and distribution of an in-house developed product with the same name (MySQL database). The software is published partly as FOSS, with a GPL license, and partly with a proprietary license, so called *double licensing*. For those users who are in need of being able to integrate the product in their own systems and re-use the source code, the licensing conditions of GPL are not applicable. Instead, such a user should have to choose the proprietary alternative. Both products are completely identical, except for the licensing conditions.

8.1.3 Added value sales associated with FOSS

Many companies that supply FOSS prepackaged also offer their own proprietary products as a complement. For example, a company can customize a complete environment for web services based on Linux and Apache, but supplemented with its own products. This combination is often cheaper than the corresponding proprietary systems and thus is of interest to the customer, who is then often offered a complete package, supplemented with suitable services.

IBM and Sun are two vendors who have chosen this model. Both have chosen Linux as an operating system and include even other FOSS-based programs in their products. Sun has also chosen GNOME for window handling for Solaris and offers of course StarOffice. In order to guarantee the continued development of these open source programs, they allow a considerable number of their own developers to contribute to the

development of the source code, which is then made available to the community. There is in this way no direct control over the development of the software itself, but instead contributions are made to the development of program code. But if the open source community does not retain a high enough standard, there is always the possibility of using the source code and carry on the development of the software on one's own as another open community, of course with a new name.

Vendors such as IBM, HP and Dell, offer Linux as an alternative operating system for their computers in order to increase sales of hardware. Oracle and many other application vendors adapt their applications to Linux. One type of application, which nowadays is almost exclusively provided for open operating systems such as Linux and xBSD, is that of firewalls. This is due to the possibility of control of source code and also because it is easy to modify the code in combination with the well-known resilience of the operating system as regards to defects and for its high degree of stability.

8.1.4 Services - training, consultation, support, operations, etc.

This is a traditional activity, which has been offered for a long time for both FOSS and proprietary software. In both cases above, with distribution of software and sales of in-house developed products, companies offer services, which are connected to the product itself. It is often these services, which generate the largest profits. Quality services are also an important criterion for many users when choosing FOSS.

But there are also many smaller companies that only offer these kinds of services. They choose some of the most successful FOSS products such as Linux, Apache, OpenOffice or product areas such as operating systems, web servers, software development, etc. and acquire professional competence in these areas and products. As FOSS becomes more and more diverse and widespread, so also will there be an increasingly expansive need for consultancy services for development, systems integration, special adaptations and training.

8.1.5 Add-ons – literature, accessories, etc.

Books, manuals, magazines, news services, etc. offer necessary and sought-after information on FOSS.

This business area is an important complement to the other business models, because knowledge about a product is a precondition for its use, further development and adaptation of the software. The O'Reilly publishing

company supports the development of FOSS by financially supporting a number of software developers. This gives the company a high degree of credibility and knowledge within the area, an investment that pays for itself through the sales of books about FOSS.

One peculiar aspect of the business area is that there is an expansive market for gadgets and items somehow connected to FOSS, e.g. clothes, mascots (e.g. the Linux mascot, the Tux penguin (coffee mugs, advertising materials, etc.

8.2 Cost comparisons

It is relatively easy to compare the costs for obtaining and upgrading licenses between free and open source software and commercial software. When considering other factors, a comparison becomes more complicated because each installation is generally the result of special conditions. If one has a well-working and stable environment – whether it is based on open source or commercial products – it is generally cheaper to remain in that environment. But as soon as changes are made, for example when upgrading an existing product to a new version, various costs arise.

8.2.1 Cost of proprietary software

Producers of proprietary software have an interest in releasing new versions of a product, even though there may or may not be a real demand or need on the part the customer. In the report from the Danish Technology Board (see below) it is stated, "the greatest competition comes from a vendor's earlier version of the same product". More often than not, a software upgrade results in a situation where even the hardware has to be replaced due to higher capacity demands on the part of the newer software. Costs due to changeover to a new product can arise when a product is taken off the market. Proprietary products oftentimes demand a steady and on-going learning curve, because every new version has new qualities, functions and characteristics.

8.2.2 Report from the Danish Board of Technology

The Danish Board of Technology has released a report: "Open Source Software – in digital public administration".³ The report contains a financial analysis of software with open source code. The report also includes a series of recommendations. One conclusion is that a changeover

³ *"Open source software – i den digitale forvaltning"*

to electronic public administration in Denmark will demand considerable investments, and in connection with this it is natural to establish which technology is to be used and who and what should direct the course of future development. Can free and open source software completely or partly replace proprietary software?

A financial model is set up in the report for evaluation of FOSS vis-à-vis proprietary software.

The report states that development costs for software are considerable, while marginal costs for copying and distribution are of less significance. What is sold is the right to use the software. A supplier with a large sales volume can set a price, which is not directly connected to the costs for development, but rather to the expected value of the right to use the software itself.

There is no capacity limit as far as production is concerned. The size of the market is the only delimiting factor for sales. Thus, there will be a tendency to establish monopolies for standard software, and competition will be difficult to maintain.

An analysis is made of the differences between FOSS and proprietary software. Initial procurement costs are compared and also running costs for maintenance. Initial financial investments in software are coupled with the right to use the product. Investment costs for FOSS are advantageous because of the right to make copies of the product. Furthermore, the user has the right to modify the software.

Maintenance costs for the two types of software are compared. For proprietary software, development and maintenance takes place under the control of the producer. FOSS is maintained in a public community.

The report presents a model for valuing investments in FOSS versus proprietary software. Investments are divided into those connected to the right to use a product versus maintenance of the product. The main portion of costs for usage rights are part of the initial investments, while financial factors related to maintenance cover a longer period of time and are more difficult to determine.

In order to facilitate reading of the conclusions of the report, a number of summarized tables from the report are used here. For more complete information, see the report itself.

8.2.2.1 Model for evaluation of costs for free and open source vs. proprietary software

		Difference between FOSS and proprietary software in free-choice situation	Difference between FOSS and proprietary software when migrating from one to the other
1	Purchasing price and/or licensing costs	Measurable difference	Measurable difference
2	User-friendliness and influence on indirect costs (e.g. long response times, unintelligible icons)	Knowledge is lacking concerning possible differences between FOSS and proprietary software. The assumption is that such expenses are more a function of design rather than dependent on type of software.	
3	Training of end users	No knowledge about this. The assumption is that this is dependent on design rather than type of software.	Cost originating in a part of lock-in effect
4	Demands for training in internal support organisation or new service agreements with vendors	It is supposed that competence building is more costly for FOSS. Usually the demand for local experts is greater for FOSS than for proprietary software. Knowledge concerning FOSS is less prevalent than for the more common proprietary products.	Cost originating in a part of lock-in effect
5	Software related technical preconditions		
5a	Surrounding software compatibility, interoperability and securing of the two factors	This cost, i.e. compatibility with third/party software, must be analysed in every individual case. Generally, one can say that FOSS is based on open standards to a higher degree, hence can be seen as more compatible.	
5b	Surrounding hardware: Technical preconditions for use, specific capacity requirements and properties along with securing of the above	Pressure to upgrade results in higher demands for resources on part of proprietary software than for FOSS	
5c	Software for maintenance and support	At the present time there are fewer options for maintenance and support for FOSS.	
6	Demands for competence in maintenance of software and the possibility of acquiring it along with operation costs either under one's own auspices or via supplier of services	Usually the demand for local experts is greater for FOSS than for proprietary software, since the vendor often supplies training and support. For FOSS competence building is based on a local initiative. The possibility to acquire services from a third party depends both on types of software and on level of demand for those services.	

7	Software operative stability, vendor's capacity to correct errors/bugs and also its strategy for the above (frequency, etc.)	FOSS has a high degree of operative stability. For proprietary software, error correction, etc. is dependent on the willingness and capacity on the part of the vendor to modify the software. For FOSS, error correction is dependent on access to developers who can make the necessary changes.
---	--	--

To illustrate the differences between FOSS and proprietary software, comparisons are made in the report between different office applications. OpenOffice/StarOffice are compared with MS Office. A general evaluation is made, see table below. Thereafter the financial model is applied to a fictitious example.

8.2.2.2 General evaluation of differences between OpenOffice/StarOffice and MS Office

		Difference between FOSS and proprietary software in free- choice situation	Difference when changing from MS Office to OpenOffice/StarOffice
1	Purchasing price and/or licensing costs	Modest licensing costs for StarOffice	Cost estimate is dependent on current licenses
2	User-friendliness and influence on indirect costs	Not part of cost estimate	
3	Demands for user training	Assumption that costs are about the same (note 1)	Limited cost due to similarities in GUI (note 3)
4	Demands for training in support/maintenance organisation	Assumption that competence building is more expensive for OpenOffice/StarOffice (note 2)	Local expertise is needed (note 2)
5	Software related technical preconditions		
5a	Compatibility and network operability	These costs can be considerable (note 4)	
5b	Surrounding hardware: Technical preconditions for use, specific capacity requirements and properties along with securing of the above	Considerable added cost when frequently upgrading MS Office	
5c	Software for maintenance and support (network management, etc.)	This cost is not included, since choice of office software is not of significance here	
6	Demands for competence in and maintenance of software	Assumption that this cost is higher for FOSS (note 5)	
7	Software operative stability and vendor's correction of errors/bugs	Costs are not included (note 1) as far as operative stability and error correction are concerned (note 6)	

8.2.2.3 Table notes

1. The working group is not aware of any study documenting the differences between OpenOffice/StarOffice concerning these points.
2. Usually, demands for local expertise are greater for OpenOffice/StarOffice than for MS Office. At the same time, there is a lower degree of general knowledge about StarOffice/OpenOffice than for MS Office due to the modest market share. Therefore there is not a great amount of consulting services and even the number of training courses is limited. The ECDL "computer driving license" is in practice based on MS Office.
3. It is a question of costs for the training itself, but also for costs that are hidden and difficult to calculate when a changeover takes place.
4. MS Office uses a closed format. Therefore, all other software producers must use reverse engineering. This involves both time and development costs.
5. This pertains in a high degree to which operating system has been installed. Office software, whether it is MS Office or OpenOffice/StarOffice, does not have much inherent capacity for central control, but is instead dependent on the underlying supportive resources of the operating system. Usually, tools are better for Microsoft Windows than for Linux, because the market is greater for Windows and therefore there are a greater number of third party vendors for Windows. Use of OpenOffice/StarOffice on a Windows platform affords the same possibilities for control as for MS Office.
6. Microsoft has frequent updates with error correction, but it is difficult to get Microsoft to include a specific, non-critical problem in their planning. For OpenOffice/Star Office, there is as yet no observation of the time it takes to correct errors with a starting point in a reasonable market share.

Based on this model a calculation has been made on the initial and running costs for an installation of general products for a workplace. The example pertains to a completely new installation of 2,000 desktop computers, servers and software. Two alternatives are shown, one with a PC as a client and one with thin clients. For Microsoft, there are two examples, one where software is upgraded at a steady pace, once every other year and which demands installation of a new desktop computer once every four years, and another example with an upgrade every sixth year.

OpenOffice does not place any great demands on the operating system and can be run on relatively old computers.

Proprietary software can also require that hardware be replaced when upgrading.

8.2.2.4 Initial costs for 2.000 workplace installations⁴

	Software in computer (PC as client)		Server based software (thin client)	
	Microsoft	FOSS	Microsoft	FOSS
Per workplace installation €	1 662	1 660	1 249	932
Total million €	3,32	2,50	2,72	1,86

Prices have been calculated on the basis of a supplier price according to Danish SKI agreement.

8.2.2.5 Yearly costs for licenses and changeover of hardware for 2.000 workplace installations

(Thousand €)	Software in computer (PC as client)			Server based software (thin client)		
	Microsoft upgrade		FOSS	Microsoft upgrade		FOSS
	After 2 yrs	After 6 yrs		After 2 yrs.	After 6 yrs.	
Upgrade	507	181	193	340	225	106
Per workplace installation	507	181	193	340	225	106
2.000 installations	1 014	537	386	679	450	213

The report shows a few examples of migration from proprietary software to FOSS. Aarhus Amt (authorities in the Danish municipality of Aarhus) has been used as an example. The administration currently has 7,000 PCs with different versions of Windows and Corel office application suite. The report contains calculations concerning what a changeover to Microsoft office or Sun's StarOffice would entail. An example shows the differences when using server-based software.

8.2.2.6 Initial costs for software, etc. plus extra server capacity – thin clients

(Million €)		MS Office XP	StarOffice	OpenOffice
Citrix		1,8	1,8	1,8
Licenses		2,62	0,20	0
Costs at changeover	Adapt templates, etc.	0,01	0,04	0,04
	Conversion programs	0	0,13	0,13
	Technical installation	0,34	0,34	0,34
	Training	0,77	0,77	0,77
	Project management	0,13	0,16	0,16
Extra server capacity		0,94	0,94	0,94

⁴ Prices here and in the following sections are roughly re-calculated to Euros

Total	6,61	4,38	4,18
€ per workplace installation	944	625	597

In cases where locally installed software is used, there is a considerably greater difference between FOSS vs. proprietary environment and substantially lower costs for the open environment.

The report also shows what savings can be attained in **the whole Danish administration** when changing over to FOSS. Below a calculation (In million Euro. Workplace installation equipment, Internet servers and advanced servers are included in the summary).

(Million €)	Short term (4 yrs)	Long term (12 yrs)
Desktop	300	220
Internet servers	130	
"Digital administration" servers	50	510
Total per year	480	730

All in all, there are potentially large savings to be made when changing over to FOSS.

8.2.3 Life cycle cost for computer networks

Cybersource Pty. Ltd. In Australia, has compared costs for Linux and Windows environments as far as total cost of ownership is concerned. For exemplification, a company with 250 employees has been used and costs for purchasing and running IT operations based on a Windows platform along with a platform based on FOSS have been studied. Two alternatives have been analysed – one where all hardware is purchased and another alternative where existing equipment is used. The calculations include costs for workplace installations, servers, connections to the Internet, business systems, networks, all standard software applications plus salaries for support personnel. Costs have been calculated for a three-year period.

In cases where there is an environment built on FOSS, the savings are calculated to about 35% over a three year period when compared to a Windows environment. This is valid when using existing hardware. When purchasing new hardware the savings are 25% for an open environment.

The differences noted apply foremostly to lower licensing costs for office applications in an open environment.

URL:

http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf

Two students at the University of Linköping, Sweden, Magnus Liljedahl and Robert Nasholm have made a similar study. In a Master's thesis from the year 2000, the two students made a financial comparison of operating systems. In order to do this, they created a life cycle model for operating systems in a network environment. The model has been used to compare total costs for Linux vs. Windows NT. The model shows that a Linux system is a cheaper alternative than Windows. The reason for the difference is, according to the authors, the higher costs for licenses for Windows.

8.2.4 Sweden

There are few comparisons made of costs for open or closed software environments in Sweden. Below is an account of a study done by The Central Authority of the Swedish Labour Market Administration (AMS).

There is no information about the use of FOSS in Swedish public administration. *Statskontoret's* blanket purchase agreement is used to a great extent in the procurement of both hardware and software. In order to provide a background for a discussion about possible savings with an increased use of FOSS, the following information is presented concerning sales using *Statskontoret's* blanket purchase agreement. The number of workplace installations in Swedish public administration is estimated to be at least 700 000.

Statskontoret's blanket purchase agreement can be utilised by state agencies and authorities, municipalities and counties. The turnover for PC blanket agreements in 2002 amounted to €30 million. This pertains to both workplace installations and so called home PCs⁵. The computers are for the most part equipped with pre-installed operating systems. The cost of an operating system is estimated to be about €160. As a rule, computers must have a pre-installed operating system in order for a supplier to receive certain rebates.

Microsoft's products are market dominators for both operating systems and office applications for workplace installations. Average cost for yearly maintenance for an installation with these products is estimated to be about €20.

⁵ A "home PC is a computer with peripheral equipment which a user can purchase through his or her employer with a rebate for the supplier and a substantial tax subsidy for the user. This has been used as a mechanism for increasing general computer literacy in Sweden.

Loaning from the Swedish National Debt Office finances computer equipment in Swedish State administration.⁶ Workplace PCs are usually written off in the space of three years. If the term of service for the equipment can be lengthened, it would mean that costs for write-offs and loans could be distributed over a longer period of time.

In Denmark, there are about 450 000 workplace installations in public administration. The socio-economic savings are calculated to be almost €480 million per year in a short term vs. a changeover to FOSS. The savings relate to both computers and software. Conditions in Sweden should be similar to those in Denmark. Recalculated for Swedish conditions with 700 000 workplace computers, the theoretical savings could be even more substantial than in Denmark.

It is not known if administrations when purchasing home PCs make a conscious effort to use competition as a price mechanism and choose among competing products when selecting operating systems and office applications.

Statskontoret only has a limited blanket agreement for database products. Therefore information is lacking concerning their purchasing value in public administration.

The Central Authority of the Swedish Labour Market Administration (AMS) has carried out a study about what a changeover from Microsoft Office to StarOffice would entail financially. Calculations show that licensing costs could be reduced substantially by migrating to StarOffice compared to upgrading to a new version of Microsoft Office. License costs for StarOffice would amount to about €0,77 million, including “internal marketing” of the product. An upgrade to a new version of Microsoft Office would cost about €2,75 million. According to AMS, Sun reckons that cost for total ownership are about €40 per user and year.

AMS has, as have other organisations, developed templates and macros based on Microsoft Office. If one upgrades to a new version of Microsoft Office or chose another office software package, these templates and macros must be adapted. AMS calculates that it would take about one day’s work per template for the changes needed. For Excel macros, it will take between two working days and up to several months for the migrating. The work needed for an upgrade to a new version of Microsoft office is reckoned to take about one tenth of the time needed for a changeover to StarOffice.

⁶ The Swedish National Debt Office is the treasury of the Swedish central government, with a mandate to manage debt, provide cash management and issue guarantees. The Debt Office is a large financial market player.

The assessment of AMS is that costs for packaging and installation of StarOffice are equal to that of Microsoft Office. The same pertains to training and support costs.

8.3 Legal aspects

This area is complicated and a more in-depth study of the legal aspects, copyright, immaterial rights, intellectual property rights, etc. for certain a type of licenses in light of Swedish legislation is recommended.

One misunderstanding concerning free and open source software is that the software is not protected by copyright. Licensing conditions, whereof GPL is one of a number of types of licenses based on copyright protection, regulates free and open source software. Mixing commercial use and distribution of free and open source software together with proprietary products is fully possible – with certain exceptions for GPL.

Furthermore, there is always the possibility of having to pay licensing fees for the product, but this rarely happens.

The problem with commercial use and distribution of FOSS together with proprietary products is also a question for discussion.

In among others, two papers: *“Open Source in a Practical Legal Perspective”* (in Swedish) by Mattias Andersson at the department of Law at the Gothenburg University School of Economics and Law and *“Legal Aspects of the Use of Open Source in Commercial and Public Activities* (in Swedish) by Olle Lindell and Ottar Kraemer at the department of Law at Uppsala University along with a number of articles by Mikael Pawlo at the Lindahl Law firm in Sweden, the legal aspects, especially relating to GPL have been covered in a very informative manner.

The IT Law Observatory of the Swedish ICT Commission plans to carry out a deeper study of the legal aspects of free and open source software during the course of 2003. The study will cover, among other things, how licenses for free and open source software relates to Swedish legislation in this area and will also look at questions concerning copyright and patentability issues.

8.3.1 General Public License – GPL

In order to fully understand what is so special about GPL and other free software licenses, one must first of all have a clear understanding of how licenses for normal software are drawn up, i.e. how current Swedish copyright laws apply to computer programs and how the standard end user licenses for software producers are usually formulated.

According to Swedish copyright laws, distribution, copying, sales, etc. are the reserved right of the author or originator of a work or a party who the author gives this right to. Buying a commercial program does not give the right to further distribution. Neither is there any right conveyed to analyse or reverse engineer the software for any other purpose than to make it able to work with other software.

Commercial software producers usually also attach a liability waiver or disclaimer to the license where the company swears itself free of all damages that can occur by using the software. This is the same procedure that is used by GPL and other licenses for FOSS.

One criticism that is often expressed concerning GPL, also among proponents of FOSS, is that it is developed for the American market.

In American copyright law there is not, for example, the forensic term Intellectual Property Right (droit moral)⁷. What does GPL contain that is so controversial? First of all, the license says that one has the right to freely copy and redistribute programs that are licensed under the GPL agreement. If the distribution takes place in a binary form, the source code must however be enclosed or made available at no extra cost. Thereafter the license stipulates that one can examine, change or modify the software, or parts of it, as much as is desired. But if redistribution takes place of the software with the modifications, whether it be in binary form or in source code, the complete modified work is licensed under GPL resulting in the new source code being open.

Some conditions, such as information concerning authors' names, etc., which would fall under the heading of intellectual property right in Sweden, must also be followed, but these are mainly of a technical nature. The GPL license ends with the following authoritative disclaimer: "*BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW.[...]*"

⁷ Entails the right to be quoted correctly, to be acknowledged as author of a work and to oppose offensive alterations. This right cannot be granted or awarded to anyone else.

One interesting paragraph in the license that hadn't needed to be included, but which heightens clarity in the reading of the license text is paragraph 5. *"You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it."* The fact that the user has no other rights than those which are expressly stipulated in the license is a subject covered in an analysis by the American law professor, [Eben Moglen](#) at the Columbia Law School.

A common misconception is that if one modifies a program licensed under GPL, then the source code must be published. This is not true, since the requirement for making source code accessible only applies to redistribution. Thus, it is fully possible for a company to modify a GPL licensed program for their own needs without having to license the changed program under GPL. But if the company wishes to sell or market their modified program, then the rules in GPL apply.

As can be seen, GPL, and other previously mentioned licenses, use copyright rules more or less as a mechanism for retaining openness in software. Those who claim that FOSS is opposed to or works against the spirit of intellectual property law are not right in their judgment. If copyright law had not existed, there would have been no protection at all against the commercialisation of open software.

Can programs, which are developed in the public sector, be licensed under GPL? The answer is yes, because the paragraph, which treats reproduction of work in the Swedish copyright law (URL), lists computer programs as one of exceptions from the public right to reproduce documents, which are drawn up or entered in official records by Swedish authorities. If this exception had not existed, GPL's demand that source code must be freely accessible would have constituted a non-compatible limitation in the right in Sweden to obtain access to public records and documents. There is also no problem using FOSS in public activities, since GPL expressly states that conditions have nothing to do with the end user, but rather regulate distribution, remodification and reproduction of the programs.

8.3.2 Comparison between different licenses

This table is partly fetched from the MITRE report "Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense".

Characteristic:	License:	GPL	LGPL	BSD & MIT	Apache	Public Domain	MS EULA
a. Can be stored on the same storage media as other types of licenses type		X	X	X	X	X	X
b. Can be used at the same time as other types of licenses		X	X	X	X	X	X
c. Can be used "above" other types of licenses		X	X	X	X	X	X
d. Can be used "below" other types of licenses		X ¹	X	X	X	X	X
e. Source code can be integrated with that of other types of licenses			X	X	X	X	
f. The user ⁶ decides if and when he wishes to publish derivative work		X ²	X	X	X	X	
g. Software can be sold for commercial purposes		X	X	X	X	X	X
h. Software can be freely copied by the user		X	X	X	X	X	
i. Software can be freely redistributed by the user		X ³	X	X	X	X	
j. Software has no restrictions concerning use		X	X	X	X	X	
k. New users can always have access to source code for derivative works		X	X ⁴				
l. New users must have full right to modify derivative works		X	X ⁴				
m. New users must have full right to distribute derivative works		X	X ⁴				
n. Software can be distributed without access to source code				X	X	X	X
o. Reused source code can be licensed under another type of license			(X) ⁵	X	X	X	
p. The original source code can be incorporated into proprietary software without a requirement that the fact be explicitly stated						X	

¹ Assuming that the different programs are completely independent of each other and can be used by themselves in other contexts.

² Assuming that the source code for the derivative work is not previously published.

³ Assuming that the source code is always distributed (or is accessible) together with the software.

⁴ Rights in LGPL do not necessarily apply to software, which only uses a program library under LGPL.

⁵ LGPL allows, in special cases, the new software to be licensed under GPL, but no other type of license.

⁶ The term user in the table above refers to an individual person, a company, an agency or authority, or an organisation.

Note that GPL expressly forbids the reuse of source code, if one licenses the software under some other type of license. However, there is nothing to prevent the use of GPL licensed software together with proprietary software, as long as they are independent.

9 Suggestions for further work

The assessment of the working group is that FOSS in many cases – both functionally and qualitatively – is fully comparable with and even better than corresponding proprietary software. FOSS should therefore be assessed side by side with proprietary solutions when procuring or purchasing software. In order to arrive at better market competition and increased interoperability between different systems, it is also necessary to place demands on open standards and open file formats when procuring software.

Many vendors of proprietary software use lock-in effects deliberately in order to complicate a changeover to alternative products. By having a strong market position and proprietary solutions, a vendor can make a situation more intricate for competing vendors and users of alternative products resulting in declining competition, higher prices and a risk of lower quality. The most advantageous situation arises when software can be chosen according to needs functionality, and not because of lock-in effects.

In order for a member of the general public to communicate with the public sector, the objective should be that no one should have to use a vendor-specific product. In achieving this ambition, an agency or authority must use open standards and formats both in development and in the procurement of their specific systems. Therefore there is a need for a procurement policy, which covers, and in certain cases places demands on open software and open standards.

One channel in future work might be to appraise the concept and ramifications of free and open source software together with decision-makers, and to discuss the cost-cutting effects which have already been ascertained along with discussing the importance of open standards and formats in the light of a more favourable foundation for competition.

An obvious next step would also be to initiate a series of pilot tests in order to evaluate the process of implementing and using alternative solutions based on FOSS

9.1 Competition related measures

The use of FOSS brings about a new type of competition, which differs from other competitors in that the product is generally not owned by any individual company and cannot therefore be “sold off the market”. There is no fee for the software itself. Instead it can be obtained free of charge on the Internet.

This means that when procuring software, an activity that usually is centred on the product itself, one must focus more on surrounding services such as support, specific service, implementation and migration. This entails a more function-oriented procurement of the software. The administration or agency should strive to find the most cost-efficient alternative, no matter if it is proprietary software or not, and that formulated requirements in a procurement situation do not discriminate against any one type of product.

Placing demands on the use of individual products and formats in a procurement situation can also be discriminating. A few well-chosen examples of this could be when an administration requires tenders and bids in MS Word format or via websites, which only work with Internet Explorer.

In order to arrive at increased competition and to avoid lock-in effects, the administration must take a number of measures (see below) such as demanding open standards, formulating a non-discriminatory procurement policy and carrying out blanket purchasing procurements which include the providing of FOSS.

9.1.1 Place demands on interoperability and open standards

In order to avoid lock-in effects and to arrive at increased interoperability, an administration must define and place demands on open standards when procuring software and when developing systems. This specifically applies to file formats for office programs and systems that include communication with the general public.

Statskontoret should therefore work for the placing of demands on open standards in procurement situations. *Statskontoret* should also cover and evaluate work being done in OASIS, work which has as its objective an open XML based file format for office software.

Another important task for an administration is communication with the general public via websites. An administration should actively support the use of open standards in these websites (according to W3C) and not discriminate against individual products.

9.1.2 Non-discriminatory procurement policy

A part of the putting into effect the objective of working against lock-in effects and supporting open standards is to produce a non-discriminatory procurement policy for software and software related services.

Statskontoret and authorities should, as soon as possible, prepare a policy for future blanket purchase procurements. One suggestion is to use the policy that has formed in Great Britain as a basis and adapt it to suit the needs and the current situation in Swedish public administration. One should, first of all, append specific requirements for the use of an open file format for office software.

For such a policy to be effectuated, a definition must first be formed concerning what open standards are applicable for public administration, for example a general and open file format for office software along with the implementation of a more consistently modular perspective in systems development including a well-defined open interface. This can be an undertaking that can be managed in the suggested organisation for a consistent and structured electronic administration, “the Swedish Government Board for Electronic Communication”.

A general procurement policy could include the following points:

- *The public administration shall consider solutions based on both FOSS and proprietary software. As basis for a decision the financially most advantageous product or solution shall be used in view of factors such as quality, function, lifecycle costs, etc.*
- *The public administration shall only use products that follow open standards in all future IT development where demands are made on interoperability.*
- *The public administration shall avoid lock-in effects when procuring products and software. Especially important in this respect are file formats for office software and for communication with other authorities and the general public.*
- *The public administration shall consider the possibility of obtaining full ownership of specially developed software or ensure that it obtains possibilities to adapt the software when procuring standard software products, whenever this is financially motivated.*
- *The public administration shall also look into the possibility of establishing a standard of using FOSS for software development that is financed by the administration itself.*

For in-house developed software to be made open, the administration must define what type of license is to be valid. There should be a need for two types of licenses, one a more liberal license, e.g. MIT or BSD, which allows

commercial interest to use the source code, and secondly a more restrictive license, e.g. GPL, which guarantees that the software remains open. More on this in the following section, 9.6, Legal aspects.

9.1.3 Framework agreement for supplying FOSS including services

At the current time, software is mostly procured in the form of off the shelf program products. The procurement processes are formulated and adapted for purchase of proprietary software, where product rather than function are put in focus. The life cycle costs for the product as far as costs for maintenance, support, service, lifetime, hardware requirements, etc., are far too seldom focussed upon.

When procuring FOSS, it is in principle, not the software itself that is acquired, but instead the provision of the same, possibly also packaged together with documentation and services.

Procurement activities to establish framework agreements for the provision of this software including services, documentation, etc., should be carried out as soon as possible in order to make it easier for the administration to choose the right product.

9.2 Catalogue of products for the administration

Since the range of various software products within different areas of use are in certain cases so extensive that an authority, such as *Statskontoret* can draw up a list of FOSS, divided into areas of use. The list will include the software, which meets demand as far as quality, function, etc. are concerned. To make it easier to find the information, the products should be accessible on a website.

Such a web based software catalogue should also be supplemented with relevant information on recommended products in order to facilitate implementation and use. The catalogue should be maintained and updated continually by the authority in charge, the suggested organisation for electronic communication. As a second step the contents could very well be supplemented with user references within the administration.

In a longer term, the web catalogue could also include a pool of software with programs that are developed and financed by the public administration according to section 9.5, Investigate in-house development of FOSS.

A special application could be the production of a complete software CD with operating system and automatic installation for desktop computers. The CD would be directly adapted to the basic needs of the administration.

9.3 Measures to improve knowledge

A very important effort for arriving at increased understanding is to inform and spread knowledge about FOSS. This can take place for example via seminars, websites with specific information for the administration and through forums for spreading of knowledge and information.

9.3.1 Spreading of information, keeping tabs on the area

Statskontoret should continue to work on information about FOSS and act as the main forum for the spreading of information and knowledge within public administration.

Statskontoret's website should include relevant information and also be updated continually.

The working group suggests that a forum be created in order to facilitate and accelerate the use of free and open source software. The target group could be IT and information strategists within the public sector, responsible for spreading and collecting information. This forum, also usable as an informal personal network, could have its own website (compare e.g. Swedish XML Academy⁸ containing information, ideas and recommendations.

9.3.2 Measures aimed at the general public

In addition to the above proposed measures for reducing lock-in effects and placing demands on open standards and formats, the project in Extremadura in Spain can also serve as a model and example.

To achieve a wider distribution and volume along with a greater amount of know-how, a public project could be started aiming at producing a public software package based on FOSS. The project should be carried out by private players and participants. The software package would be distributed on a ready to install CD and could also be available from public libraries for private use. The main objective is to provide tried, tested and free

⁸ www.xmlakademin.nu

alternative solutions based on FOSS to the general public for the purpose of communication with authorities along with, of course, easy to use and software of the same quality for access to 24/7 agencies within e-government.

9.3.3 Modifying ECDL – European Computer Driving License

The ECDL is a certificate showing that the owner has knowledge of the most common concepts in the area of IT, can use a PC and is familiar with standard areas of use and applications for a personal computer.

The ECDL is a certification of knowledge and competence, and is based on know-how requirements, which are specified in a document administered in Sweden by the Swedish Information Processing Society. ECDL is an internationally accepted certificate. The overall aim of the ECDL is to heighten the basic level of competence in the area of IT and to increase people's ability to use a personal computer and to be able to manage the more common applications. The target groups are companies, authorities and private individuals.

ECDL comprises the following module tests:

The ECDL consists of a seven module tests, which lead to the qualification. The modules are:

- Basic concepts of IT
- Using the computer and managing files, operating system
- Word processing
- Spreadsheets
- Database
- Presentation
- Information and communication, Internet and e-mail

At the current time, the ECDL is adapted to and designed for proprietary software.

An adaptation of the most common FOSS applications, especially for PCs, is therefore of the greatest importance in order to avoid lock-in and discrimination against products. This adaptation will be especially essential when public administration begins to use FOSS to a greater extent on its desktop computers.

For example, an adaptation should be carried out as soon as possible for FOSS products such as GNU/Linux, including windowing systems such as GNOME and KDE, OpenOffice/StarOffice and MySQL.

9.3.4 Seminars

Competence raising measures, such as seminars, geared to the needs of an administration, should be planned and carried out as soon as possible; it is essential to treat the area of FOSS matter-of-factly and to clarify the differences between freeware and shareware.

9.3.5 Forum for spreading of knowledge

Statskontoret should also allocate resources at an overall level for support and help to authorities in procurement situations and how to apply recommendations. *Statskontoret* should also function in an advisory and competence spreading capacity when implementing and using FOSS.

9.4 Pilot installations

At the current time there are no cases of any administration migrating from a traditional proprietary solution to one based on FOSS. An obvious next step would therefore be to initiate a series of pilot installations of FOSS in a middle-sized authority, organisation or department.

The pilot installations should include different areas of use, for example workplace computers, web servers and databases. When starting out on these pilot tests it is especially important to document the migration process itself and also the problems encountered by extricating oneself from a potential lock-in situation. Also the financial aspects must be studied. The Swedish National Labour Market Administration, the Swedish national Police Board and the Swedish National Road Administration have shown interest in participating in a pilot scheme.

For PCs and workplace computers, it is mostly interesting to test alternative office software, but also a complete standard solution for the desktop med Linux as an operating system, office programs, e-mail client and web browser.

Another type of pilot could include a newly started administration or agency without any technical legacy to have to take into consideration. In this case the task would be to find an optimal software environment based on a combination of selected FOSS and proprietary applications.

Another conceivable step would be to start a trial project with FOSS especially for municipalities in for example workflow and information management. This could be carried out in both a large and a small scale. A study could be made of how FOSS in a developing environment works in such context or carry out a case study at a municipality which introduces IT systems based on FOSS.

An interesting project would be to supply schools and universities with a standard solution based on FOSS, preferably in collaboration with corresponding projects in Denmark ([GnuSkole](#)) and Norway ([Linux i Skolen](#)).

9.5 Investigate in-house development of FOSS

Besides purchasing standard software, authorities in Sweden develop their own specially adapted applications for their own use. This can be done with their own resources, but it is more usual for an authority to procure these services from a vendor or hire consultants to develop the software for them. Many administrations purchase, often independent of each other, development and adaptation of similar systems, e.g. systems for document and information management, accounting software, personnel and payroll software, websites and security related systems such as firewalls and PKI-solutions.

There ought to be a great potential for savings by cooperation in the procurement process, in development and by making in-house developed software accessible for other parts of public administration or even for the general public or reuse.

In tune with the recommendations for a new procurement policy, public administration should therefore study the possibility of using as a standard FOSS for software development financed by the administration. Public administration should also strive to attain a situation where in-house or procured development of software results in the use of FOSS, accessible for others to reuse!

In order to put this into practice, there is a need for both a survey of the scope and type of in-house developed software and a central coordination of the software, which is produced, and which will be developed. There is also a need to define license types for in-house developed software (see next section) and furthermore to look over the legislature (Law on Public Procurement (LOU)) pertaining to how in-house developed software can be made available to other agencies and authorities.

The responsibility for coordination and distribution should be vested in a central authority, possibly in the new "coordination authority". A web based register of software included in the software pool should also be set up.

A part of the work, with the aim of sharing in-house developed software, should also be to cover the work being done in the EU project for creating a software pool within the European Union (POSS – Pooling Open Source Software).

9.6 Legal aspects

By only using FOSS there is no great need to delve into the legal intricacies of certain types of licenses, such as GPL. It is only when distributing derivative works that this type of license places specific demands on non-dependence and openness.

It is among other things the need of a European adjustment to the GPL license and a closer study of the disclaimers or liability waivers that exist along with liabilities themselves as related to Swedish (and European) law.

In order for public administration in Sweden to publish and distribute in-house developed software as FOSS, it is necessary that a more careful study of the problems involved be undertaken.

Most probably there will be a need for two different types of licenses:

- A more liberal license, e.g. MIT or BSD, which allows commercial interests to use the source code - this is in order to promote development and cooperation with private interests.
- A more restrictive license, e.g. GPL, which guarantees that the software remains open and cannot be reused commercially. This is especially valid for areas that can be seen as a part of a general public IT infrastructure.

Both alternatives have their advantages and disadvantages, but bring about a clear improvement when compared to the current accepted procedures, where different authorities risk having to pay two-fold or more for similar software and where the general public risks having to first contribute via tax revenue for developing software and then having to pay for it again to use it for their own purposes.

Which type of license is suitable for different situations must be decided from case to case.

A continued and more detailed study of the legal aspects of using FOSS licenses will be carried out in the spring of 2003 by the IT Law Observatory of the Swedish ICT Commission. *Statskontoret* should cover and follow up this work so that it can be integrated with other future undertakings.

10 References

10.1 Bibliography

“**A Framework for Open Source Projects**” - Gregor J. Rothfuss, Zurich, November 12, 2002 – http://greg.abstrakt.ch/docs/OSP_framework.pdf

”**A Guide for Small and Medium Enterprises**”, a pamphlet from the German ministry on the subject of FOSS. The Nordic Industrial Fund has commissioned the translation to English: <http://oss-broschuere.berlios.de/broschuere/broschuere-en.html>

“**Free/Libre and Open Source Software (FLOSS): Survey and Study**”, EU-report, June 2002. <http://www.infonomics.nl/FLOSS>

Free Software Foundation, organisation founded by Richard M. Stallman for supporting development and spreading of information of the GNU-project. <http://www.fsf.org>

“**Free Software Matters**”, a number of articles on FOSS by Eben Moglen, Professor of Law & Legal History, Columbia Law School. <http://emoglen.law.columbia.edu/>

Gnuheter, website with news on FOSS, created by Mikael Pawlo and Patrik Wallstrom. <http://www.gnuheter.com> (*In Swedish only*)

GNU-project, contains among other things information on projects and the GPL-licenses. <http://www.gnu.org>

”**Life Cycle Cost for datanätverk - en ekonomisk jämförelse av operativsystem**”, Master’s thesis by Magnus Liljedahl & Robert Nasholm, Department of Economics, Linköping University, Sweden, 2000. (*In Swedish only*)
<http://www.cendio.se/page.news.paper-lcc>

”**Linux vs. Windows – Total Cost of Ownership**”, a comparison of total costs between systems based on Linux/Open source and MS Windows, by Cybersource Pty. Ltd. 2002.
http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf

”**Nagot om fri programvara och öppen källkod – nya licenstyper för datorprogram**”, by Mikael Pawlo. <http://harvard.pawlo.com/pawlo-nir.pdf> (*In Swedish only*)

NERA (National Economic Research Associates), “Government Preferences for Promoting Open-Source Software: A Solution in Search of a Problem”, May 2002. http://ssrn.com/abstract_id=313202

OASIS: Organization for the Advancement of Structured Information Standards. Carries out work for among other things producing an XML-open file format for office software. <http://www.oasis-open.org/>

Open Source Initiative: Organization for the advancement of FOSS. Also certifies licenses according to the Open Source Definition. <http://www.opensource.org>

”Open source software – i den digitale forvaltning” (Open Source Software in Digital Public Administration), Report from the Danish Board of Technology, an analysis of FOSS in public administration, 2002. (*In Danish only*)
<http://www.tekno.dk/subpage.php3?article=851&survey=1&language=dk&front=1>

”Open Source Software – The use of Linux and other open source software in national administration” Norwegian *Statskonsult’s* report on whether or not FOSS can contribute to vendor independency and cost-cutting in national administration, 2001.
<http://www.statskonsult.no/publik/publikasjoner/2001-07/r2001-07eng.pdf>

”Open Source – Ur ett praktiskt juridiskt perspektiv”, term paper by Mattias Andersson, Department of Law, School of Economics and Commercial Law at the University of Gothenburg, Sweden, 2000.
<http://www.itkommissionen.se/index.html> (*In Swedish only*)

”Oppen kallkod och offentlig sektor”: Term paper by Charlotte Kihlstrom and Mattias Kihlstrom, Blekinge Institute of Technology. (*In Swedish only*)
<http://www.kihlstrom.com/oppenkallkodochoffentligsektor/>

”Pooling Open Source Software (POSS) Feasibility Study”, June 2002, describes a possible model for how the user within the EU can manage and exchange in-house developed software with other users.
<http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&documentID=625&parent=chapter&preChapterID=0-17-134>

”Report on the suitability of the OpenOffice.org office suite and the Linux operation system as the workstation standard of the City of Turku”, performed by the IT Department of the City of Turku, 2001.
<http://www.turku.fi/tieto/liite44.rtf>

The Center for Open Source in Government - EGovOS.org, organizes among other things conferences on Government and has published a report from MITRE: **"Use of Free and Open Source Software in the US Dept. of Defense"**.

<http://www.egovos.org/>

"Why Open Source Software / Free Software (OSS/FS)? Look at the Numbers!" David A. Wheeler, November 2002. The document presents factual information in the form of statistics on both the current and expected use of FOSS. http://www.dwheeler.com/oss_fs_why.html.

10.2 Other literature

Code and Other Laws of Cyberspace, Lawrence Lessig, Basic Books, Aug 2000, ISBN 0-465-03913-8.

Free Software, Free Society: Selected Essays of Richard M. Stallman, Oct 2002, GNU Press, ISBN 1-882114-98-1.

The Business and Economics of Linux and Open Source, by Martin Fink, Sep 2002, Paperback. Prentice-Hall, New Jersey, UK, 2002 ISBN 0-130-47677-3).

The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary, Eric S. Raymond, O'Reilly & Associates, Farnham, UK Feb 2001, ISBN 0-596-00108-8.

“The Future of Ideas: The Fate of the Commons in a Connected World”, Lawrence Lessig, Vintage Books, USA, Nov 2002, ISBN 0-375-72644-6

"Understanding open source software development", Josef Feller & Brian Fitzgerald, Addison-Wesley 2002, ISBN 0-201-73496-6.