

Open Source Software (OSS) in today's Information and Communications Technology (ICT) sector

Abstract

In today's ICT sector the use of Open Source Software is in very prevalent use, much more than the average technology user may know or suspect. This white paper provides an overview of Open Source Software, including its definition, licensing arrangements, current penetration, its capability, its pros and cons and the general industry/cultural acceptance of its use.

John Sheehy

1 Executive Summary

Open source software (OSS) is defined as computer software for which the source code and certain other rights normally reserved for copyright holders are provided under a software license that meets the Open Source Definition or that is in the public domain. This permits users to use, change, and improve the software, and to redistribute it in modified or unmodified forms. It is very often developed in a public, collaborative manner. The formal definition of open source has been fully developed by the Open Source Initiative (OSI). The definition consists of ten key attributes. The OSI is also the keeper of open source activities across the industry and promote how Open Source Technologies, licenses, and models of development can provide economic and strategic advantages.

Open Source Technologies can be considered as the ultimate disruptive technology within the ICT, given its growing market share of 6% of an estimated trillion dollars IT annual budget. The main driver of open source is cost reduction specifically in relation to cost of acquisition, given that open source is in the main free when compared to propriety products, i.e. no licence fees. While the acquisition cost is low, the Total Cost of Ownership is not free, when installation, support, maintenance and customisation are considered.

There are two types of OSS licence, i.e. General Public License (GPL) and Lesser General Public License (LGPL). The GPL grants the recipients of a computer program the rights to use and modify software and ensures that these freedoms are preserved, even when the work is changed or added to. The Lesser General Public License (LGPL) is a modified version of the GPL. The LGPL grants rights to the program itself but does not apply these restrictions to other software that merely links with the program. The LGPL is primarily used for software libraries.

The adoption of OSS across the ICT industry varies considerably across the sectors within the industry and the application/platform areas where open source is deployed. In the context of the application/platform areas, open source has had more acceptance in the backend environment than in the frontend environment, i.e. in the area of server support rather than clients and desktop PCs. This is in the main associated with the perceived capability of OSS products. The products tend to be more accepted in the back end where standard interfaces, open formats, scalability, performance and security are significant drivers. OSS does have the perception that its human interface capability and features are not as strong as propriety products, specifically Microsoft Windows and Office products. To an extent this is driven by the fact that Microsoft has 87% of the desktop PC market. Having said this, the perception of the features richness and usability of OSS is changing in a positive direction, when taken from an OSS perspective.

When one looks at the various sectors within the industry, i.e. business, consumer and governments, the acceptance of OSS varies considerably. In the business environment, OSS has had significant success, with its adoption in the financial, e-trading, entertainment and communications industries. Particular successes included the adoption of OSS based servers, specifically Linux with approximately 23% of the market and the Apache web sever with 66% market share. It is also interesting to note that in the communications sector, open source management products are beginning to have an impact. The leaders can be considered openNMS and Nagios. In the consumer environment, specifically PCs, the penetration of open source products has had limited success, due in the main to the dominance of Microsoft products. As PCs become commodity product this situation is beginning to change, since software from Microsoft has become a significant cost factor in a PC and OSS is seen as a

solution to addressing this cost. Governments are also beginning to take note of OSS, largely driven by cost, with several countries beginning to mandate the use of OSS, as is the case with the German federal government. In the UK, the government's OSS policy promotes a 'level playing field' in which OSS solutions should be considered alongside proprietary ones in IT procurements. While the use of OSS at the federal level in the United States is not as enlightened as other countries, the new administration have requested leading technology industry opinion leaders to report on the benefits of the use of open source.

Like everything OSS is not a panacea of solutions to all problems in the ICT industry. It has its pros and cons. It is fair to say that its major pro is the fact that it is free in the main. But if one looks at other areas it does have negatives as well as positives. There is a perception that it is less feature rich than propriety packages, but this situation is changing. The total cost of ownership is not zero, when deployment, maintenance and support are considered. In terms of its capability including scalability, performance, reliability, and security it is as strong if not stronger than competing propriety packages. In general OSS is easier to deploy, given that no licence management is required, but maintenance is not as mature as that which is offered by commercial software organisations. Having said this, companies are stepping in to fill this area, e.g. RedHat's support of Linux.

In the past there has been a perception that OSS is close to anarchy. This is not true, many established high profile companies are thriving by using and selling open source software. Open source represents a paradigm shift for the technology industry. OSS does and may undermine some existing businesses. There's little reason to buy a proprietary program when a comparable open source alternative exists. However, the software industry was already transforming from a product model to a service model. In the Telco space, Bull is a particular champion of open source, defining themselves as 'Architect of an Open World'TM. Bull creates open source components, develop existing open source components to deliver added value and act as an open source integrator. HP, through its Open Source Integrated Portfolio (OSIP) technologies, offers software, middleware, services, operating systems, and platforms. Red Hat sells open source software with value added, including service.

Open source is here to stay as it has reached a critical mass in its deployment. At this point, it is reasonable to assume that OSS will continue to permeate through the ICT industry, with its overpowering acquisition cost advantage, especially given the ongoing requirement to reduce product cost.

2 Glossary of Terms and Abbreviations

Adware	Adware or advertising-supported software is any software application in which advertisements are displayed while the program is running. These applications include additional code that displays the ads in pop-up windows or through a bar that appears on a computer screen.
Bugs	An unwanted and unintended property of a program or piece of hardware, especially one that causes it to malfunction. E.g. "There's a bug in the editor: it writes things out backward". The identification and removal of bugs in a program is called "debugging".
Commercial Software	Commercial software is software developed by businesses which aim to make money from its use. Most commercial software is proprietary, but there is commercial free software and there is non-commercial non-free software.
Demo Software	Commercial software that is available for free download and use but with limitations, e.g. certain features are disabled, or the software works only for a limited trial period.
FLOSS	Acronym for Free (Libre) Open Source Software .
FOSS	Acronym for Free/Open Source Software .
Free software	Free software is defined by the users' prescribed freedom to run, copy, distribute, study, change and improve the software. By definition free software is open source.
Freeware	The term "freeware" has no clear accepted definition, but it is commonly used for packages which permit redistribution but not modification (and their source code is not available).
General Public License	General Public License: a specific set of distribution terms for free software that specifically does not let redistributors add any additional restrictions when they redistribute or modify the software. This means that every copy of the software, even if it has been modified, must be free software.
GPL	See General Public License .
Linux	A trademark name for an open source version of the UNIX operating system. A free operating system used extensively for running servers and increasingly desktop PCs.
NGO	Non-governmental organization.
Open Source Software	Software released under a license which gives users the right to both use the software and modify the underlying source code.
Operating System	The software which handles basic computer functionality such as the user interface, hardware and file storage. Popular operating systems include Microsoft Windows, Macintosh operating system and Linux.
OSS	See Open Source Software

PC	Personal computer.
Proprietary Software	Proprietary means that some individual or company holds the exclusive copyright on a piece of software, at the same time denying other people access to the software's source code and the right to copy, modify and study the software.
Public domain	The term "public domain" is used to describe publications, software, and other resources which are not protected by copyrights or patents.
Semi-Free Software	Semi-free software is software that is not free, but comes with permission for individuals to use, copy, distribute, and modify (including distribution of modified versions) for non-profit purposes.
Shareware	Shareware is software which comes with permission for people to redistribute copies, but says that anyone who continues to use a copy is required to pay a license fee.
Software License	A software license sets out the terms under which the software may be used, and serves as an agreement between the producer and the users of the programme.
Source code	The form in which a computer program is written in a programming language such as C++ and Java.
TCO	See Total Cost of Ownership .
Total Cost of Ownership	The sum of all expenses directly related to the ownership and use of a product over a give period of time, usually a few years. For software, these expenses can include license fees, upgrade expenses, required hardware purchases, staff training, migration time, changes in staff efficiency, vendor management and support fees.
Usability	Usability refers to how easily and efficiently a programme allows users to carry out their required tasks. Bad usability can result in user frustration, wasted time and mistakes.
Vendor lock-in	Vendor lock-in is a situation in which a customer is dependent on a vendor for products and services and cannot move to another vendor without substantial costs. It is often used in the computer industry to denote the lack of compatibility between different systems which intentionally or unintentionally forces a customer to continue to use products and services from a particular vendor.

3 Introduction -

Over the past 25 years approximately, Open Source Software (OSS) has become increasing prevalent and utilized in the Information and Communications Industry (ICT). While to

movement to OSS has its roots back in the 1960s in open standards, specifically during the development of the Advanced Research Projects Agency Network (ARPANET), which eventually led to the birth of the Internet in the late Sixties. During the mid-Eighties it gathered momentum through the free software movement, but the OSS label came out of a strategy session held in reaction to Netscape's January 1998 announcement of a source code release for Navigator. This session used the opportunity before the release of Navigator's source code to clarify a potential confusion caused by the ambiguity of the word "free" in English. It is generally accepted that 'open source' movement is thought to have begun from this point, but it is important to note that software which now falls under the umbrella of 'open source' was available prior to this point.

Open source software (OSS) is defined as computer software that has its underlying 'source-code' and certain other rights normally reserved for copyright holders made available under a licence. This permits users to use, change, and improve the software, and to redistribute it in modified or unmodified forms. It is very often developed in a public, collaborative manner. While this is a generic definition, in fact a very specific definition has been generated by the Open Source Initiative, (Ref 1), which will be provided later in the white paper.

Initially OSS was frowned upon by the international business community and governments, it has now reached a market penetration when it has in certain applications reached almost 50% market share. As a result the corporate community and governments have developed policies for its use and have begun to embrace it.

This White Paper provides an overview of OSS, its current market and utilisation, its capabilities, its pros and cons and the impacts on business, academia and the community at large. While the white paper addresses OSS in general, there is specific mention of the use of OSS in the communications environment. The specific sections of the white paper are the following –

- **Executive Overview –**

Provides an executive overview of the white paper, with specific focus on the key messages and findings, as outlined at the beginning of this document.

- **What is Open Source Software –**

Provides a formal definition of Open Source Software and how it is controlled.

- **Licensing –**

Provides an overview of the types of OSS licences that are defined.

- **Market penetration of OSS –**

This looks at the overall penetration of OSS and provides some detail on specific success stories. The areas addressed include business, consumers and governments.

- **Pros and Cons of OSS –**

This section looks at the pros and cons of OSS compared to commercially provided software, with specific focus on areas such as the cost of ownership, features and quality, users and migration, etc.

- **The Future of OSS –**

This section looks at the possible future trends that may develop through the use of OSS.

4 What is Open Source Software -

It was noted in the introduction that OSS is computer software that has its underlying 'source-code' made available under a licence, which can allow developers and users to adapt and improve it. There is a very specific definition provided by the Open Source Initiative (OSI). The OSI are the stewards of the Open Source Definition (OSD) and the recognized body for reviewing and approving licenses as OSD-conformant. The specific definition developed by the OSI is the following (Ref. 2):

1. **Free Redistribution**

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

2. **Source Code**

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

3. **Derived Works**

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

4. **Integrity of the Author's Source Code**

The license may restrict source-code from being distributed in modified form *only* if the license allows the distribution of "patch files" with the source code for the purpose of modifying the program at build time. The license must explicitly permit distribution of software built from modified source code. The license may require derived works to carry a different name or version number from the original software.

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

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

6. **Discrimination Against Fields of Endeavour**

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

7. Distribution of License

The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties.

8. License Must Not Be Specific to a Product

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

9. License Must Not Restrict Other Software

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

10. License Must Be Technology-Neutral

No provision of the license may be predicated on any individual technology or style of interface.

While the OSI provides the overall definition of OSS, specific organisations have been setup to define the technical capability, functionality and usages of specific OSS products. Specific examples of these include the Apache Software Foundation (Ref 3) and the Mozilla Foundation (Ref 4). The Apache Software Foundation provides support for the Apache community of open-source software projects, including the ubiquitous Apache Web Server and many other web application tools. The Mozilla foundation provides support for the Firefox browser, which has become the browser with the second highest market share after Internet Explorer. Both of these organisations have embraced the OSS concept.

In the area of communications technologies, Linux is fast becoming a significant player in the OS space. Linux is an OSS package, which is controlled by the linux.org, with the active support of Linus Torvalds, the creator of Linux. In the area of Network Management, openNMS is an enterprise grade network management platform developed under the open source model. It consists of a community supported open-source project as well as a commercial services, training, and support organization. The capability of the product includes:

- High performance - A single instance of openNMS supports monitoring of a large number of nodes from a single server.
- Automation - Minimizes the amount of manual configuration and reconfiguration needed by automatically performing tasks on a regular basis, like discovering new nodes, detecting services on new and existing nodes, and gathering response time and performance data.
- Rule-based configuration - Flexible rules can be used to specify what services are polled on certain devices, which SNMP data is collected and how often, and to whom various

notifications should be delivered, without having to explicitly identify and configure each host in the monitoring system.

5 Licensing –

The following describes the different types of open source licences:

5.1 GPL (General Public License)

The GPL grants the recipients of a computer program the rights to use and modify software and ensures that these freedoms are preserved, even when the work is changed or added to. This also applies to software that is linked to a GPL library, i.e. the linked software must have a GPL license.

5.2 LGPL (Lesser General Public License) or GPL for library

The Lesser General Public License (LGPL) is a modified version of the GPL. The LGPL grants rights to the program itself but does not apply these restrictions to other software that merely links with the program. The LGPL is primarily used for software libraries. In this case if the software is linked to LGPL library, the linked software does not have to be controlled under a GPL license.

5.3 BSD (Berkeley Software Distribution) license

Under the BSD license, software can be freely distributed, modified and used in both an opensource and proprietary situation with some key restrictions. The restrictions placed on users of software released are that if they redistribute such software in any form, with or without modification, they must include in the redistribution (1) the original copyright notice, (2) a list of two simple restrictions and (3) a disclaimer of liability. The restrictions can be summarized as (1) one should not claim that they wrote the software if they did not write it and (2) one should not sue the developer if the software does not function as expected or as desired. Some BSD licenses additionally include a clause that restricts the use of the name of the project (or the names of its contributors) for endorsing or promoting *derivative works*.

5.4 Impact on company or product IP

Code is not distributed to any other company as long as the software is not distributed. The organisation can keep the development secret. If the software is distributed under GPL and someone else changes something in the software and distributes the new version, the code has to be provided with it. This means that the originator has access to the improvements.

6 Market penetration of Open Source Software –

OSS is becoming an increasing player in the software market, as outlined by the following two statements from respected analysts:

"Open source will grab at least 22 percent of the software market in the next five years, due in part to increased adoption by small to medium-sized businesses looking to save money. The increased adoption will most likely occur in application development and integration tools. And if additional adoption occurs in the open source customer relationship management and enterprise resource management spaces, the takeaway will be even larger." (Ref.5)

"Open Source software is raising havoc throughout the software market. It is the ultimate in disruptive technology, and while to it is only 6% of an estimated trillion dollars IT budgeted annually, it represents a real loss of \$60 billion in annual revenues to software companies." (Ref.6)

To break down the penetration of OSS, it is worth considering three different segments. These are:

- Business.
- Consumers.
- Governments.

6.1 Business

The drivers for business to adopt open source are in the main cost and to a lesser extent strategic, as the world economic situation worsens and their trading models evolve. There are many examples of large corporations adopting the open source paradigm.

In the area of cost reduction several large financial companies have adopted open source, these include Merrill Lynch who have adopted a large scale Linux deployment in an effort to reduce costs. They in turn are being followed by Credit Suisse and Goldman Sachs. In the area of e-trading Amazon have chosen open source. They allege that they have saved \$17 in technology and communication costs per quarter. Amazon also feature a significant number of open source applications in the Amazon Elastic Compute Cloud (Amazon EC2)

As trading models have grown more dependent on the Internet, companies are using more open source software. Typically the use of open source products and applications has been confined to the backend platform, such as web servers using Apache and the operating systems on these servers, i.e. Linux. The reputation of open source being less prone to viruses and more scalable is also a significant factor in its adoption.

The two classic examples of the adoption of open source packages are the Apache web server and the Linux operating system. In the case of Apache it holds a market leadership position with 106 Million sites, representing approximately 46 % of the market. It has a 47% lead over its closest competitor, Microsoft-IIS, much greater than on the web as a whole. When

consideration is given to the web servers used by the million busiest websites Apache is the clear leader with a 66% share. (Ref. 7, Fig. 1)

Server Share amongst the Million Busiest Sites, March 2009

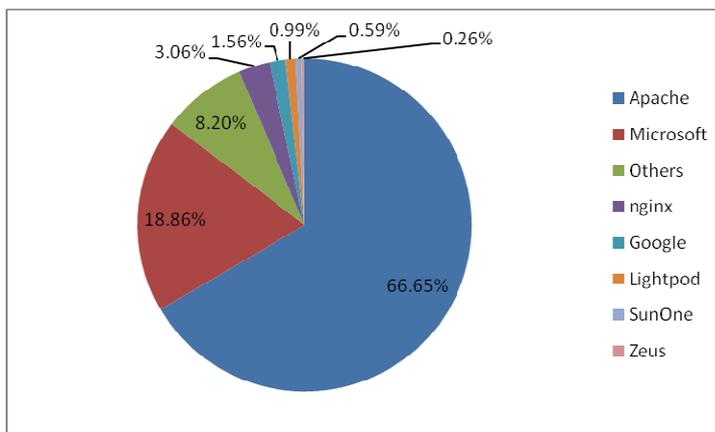
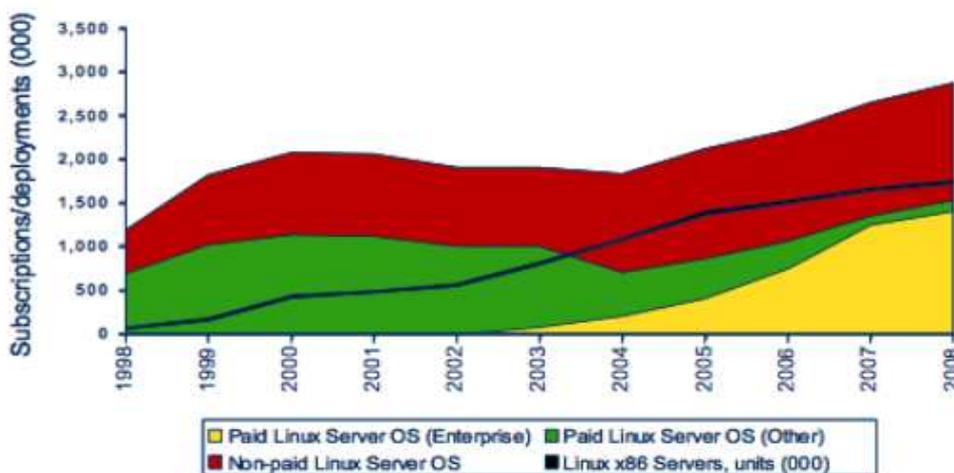


Figure 1.

In the case of Linux the market has consistently grow over the past several years. It's important to recognize that not all of Linux's success can be seen in corporate financial results. Much of the growth of Linux comes from unpaid deployments, which continue to account for a healthy margin of total deployments (Ref. Fig. 2):

Linux Server and Operating System Growth, 1998-2008



Source: IDC, 2008

Figure 2.

Linux is set to outpace the larger market, with customer spending on Linux expected to grow year over year by 21 percent in 2009. The larger software market, meanwhile, will struggle to deliver 2 percent growth in 2009. And from 2008 to 2013, the Linux market is set to grow \$12.3 billion to \$35.5 billion, representing a 23.6 percent compound annual growth rate.

Even so, it is important to note that the size of Linux versus, for example, Windows, is telling: the Microsoft software ecosystem was \$149 billion in 2008. IDC rightly points out that "even with a sub-10 percent growth rate through 2013, (the Microsoft ecosystem) will add \$56 billion in spending.

Virtualization is expected to be a big driver of Linux. While cloud computing is also expected to drive Linux and open-source adoption, the real money is coming from increases in Linux adoption--from 13 percent to 18.6 percent--for more traditional workloads like ERP, database, etc. Most of this growth in traditional workloads is coming at Unix's expense. (Ref. 8)

In the communications sector, the use of open source server platforms is consistent with the general industry penetration. It has not been as prevalent in the applications area. This is beginning to change with the introduction of the openNMS, which is an Enterprise and Carrier grade network management platform developed under the open source model, (Ref. Fig. 3). The product is written in Java and packaged for Windows, Linux and several other Unix distributions. The main marketing proposition of openNMS is that it allows an organisation to get rid of the software licensing cost associated with propriety solutions and as it is more flexible it can also reduce deployment time while providing more custom functionality.

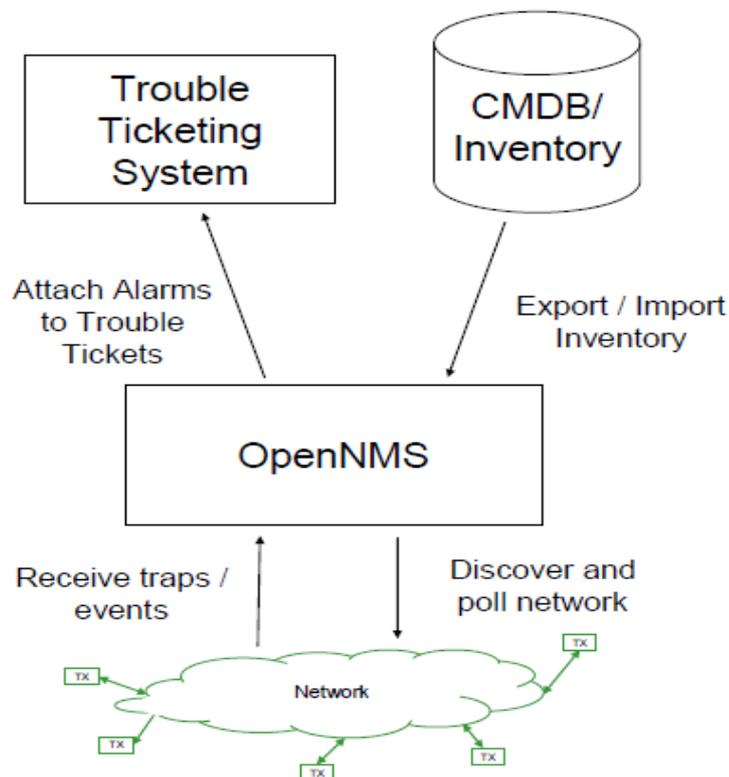


Figure 3 - Open EMS in Context (Ref 9)

While openNMS is still in its early stages, it has some significant reference sites including the BBC and Wind Telecomm in Italy. The openEMS group is also involved with the Telemangement Forum in support of the TM Forum Interface Program (TIP) which is responsible for all the TM Form work on interface specifications.

6.2 Consumers

In the consumer market, there have been mixed results in the penetration of open source products. This has largely been due to market dominance, specifically in the PC desktop environment, where Microsoft holds sway. The other reason for this is that the desktop user frontend requirements are considerably different to the requirements of the backend user. Typically the frontend user is focussed on GUI functionality and ease of use, while scalability performance and reliability take a secondary place.

As a result Linux has little penetration in the desktop OS environment, with 1.0% share, whereas Microsoft has 87%, Apple has 10% and others stand at 2%. Having said this, the Linux, Apple and others share is increasing at the cost of Microsoft's share. As the PC becomes a commodity there is strong competition to reduce costs. As more users choose unbranded PCs, prices drop. However, proprietary software like Microsoft Windows and Office has a virtual monopoly in the PC market. The cost of the software determines the lowest possible price of a Microsoft Windows PC, and that cost is steady. (Ref. 10, Fig 4)

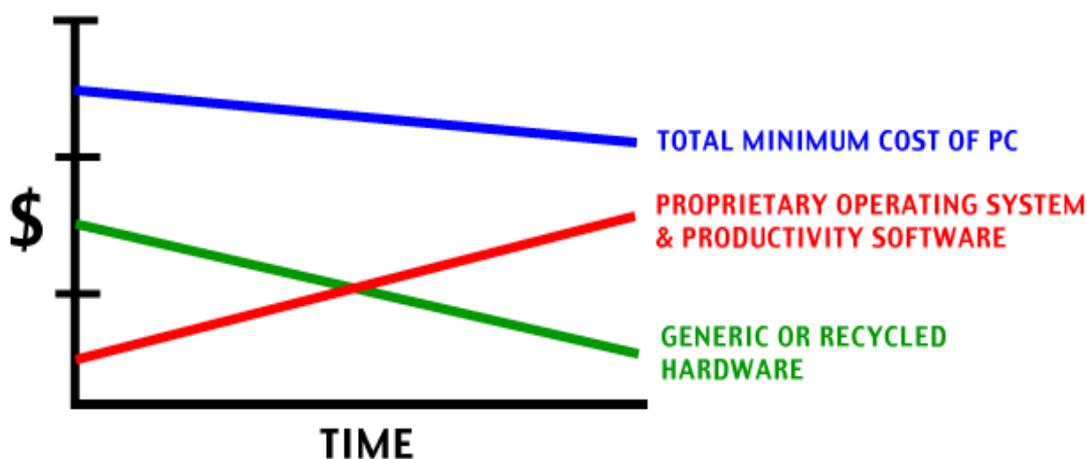


Figure 4 – Cost Drivers

The use of open source software in PCs is providing a significant cost reduction, reference the significant reduced costs of netbook PCs, which use Linux and open source application packages. Companies such as Dell and HP-Compaq are already deploying Linux in some of their PC ranges.

6.3 Governments

Governments' interest in OSS is increasing, due to their reliance on sophisticated software and cost. For example the UK Office of Government Commerce released a series of case studies outlining how OSS has been used in the public sector. However, UK parliamentary responses to questions on the use of OSS in government show that uptake is still limited. The Office of the Deputy Prime Minister is funding the 'Open Source Academy' project. This is intended to overcome barriers to uptake of OSS in local government such as lack of information, skills, confidence and lack of suitable products. From a UK government perspective the key issues associated with open source are:

- Acceptance of open source software is increasing in both the public and private sector. It is now accepted that open source is a viable and credible alternative to proprietary software for infrastructure and for most desktop users.
- The government's OSS policy promotes a 'level playing field' in which OSS solutions should be considered alongside proprietary ones in IT procurements.

In Germany, the government is deploying Linux and the open source model at the federal, state and community levels. It argues that open source is more secure than proprietary software and explains that Germany is seeking independence from any one vendor, while "creating more diversity in the computer field.

In the United States, the situation is more complex and disjointed. While several federal agencies, including the Department of Defence use open source to a greater or lesser (usually lesser) there no federal policy on the use of open source. As a result of the recent change in administration, leading technology industry opinion leaders, including Scott McNealy the Sun CEO, have been asked to report on the benefits of the use of open source. The factors being analysed include quality of products, costs, reliability, etc.

7 Proc & Cons -

The following outlines some of the pros and cons associate with open source packages, (Ref. 10).

7.1 Total cost of ownership

<i>Open source has a much lower price.</i>	True
<i>The total cost of open source is lower.</i>	Not Proven
<i>Some software isn't compatible with open source.</i>	True

Table 1 – Total Cost of Ownership

The price of an open source program is usually far less than a comparable proprietary program, *Free* is a very good price. Open source software doesn't have to be "no fee" but most programs are. Users can either download the software directly or pay a negligible fee to have a CD-ROM burned and shipped.

The other costs of open source software are potentially lower. For instance, the software typically has lower hardware requirements than proprietary alternatives. A specific solution may need fewer, older computers. Solutions usually don't depend on proprietary hardware. Software can be downloaded from the Internet quickly and conveniently.

It's not clear whether the TCO of open source is really lower. The salient issue is expertise. The cost and availability of open source expertise relative to proprietary expertise (e.g. Cisco Certified Experts) is open to debate. In open source solutions, the quality and availability of help is proportional to interest and use. Currently, there are more proprietary packages that include support and written guarantees. Having said this, some companies are starting to offer comparable packages for open source solutions.

Choosing any solution may deny interface other software. This so-called opportunity cost may not be apparent for years, when the need for the other software emerges. In many cases, open source is still the minority solution. For example, the number of Linux desktops is low compared to Microsoft Windows. By choosing a Linux desktop, a user forecloses on some software because it may never be created for or ported to Linux. Likewise in the area of network management, some propriety operations support tools may not be compatible with open EMS.

7.2 Features

Open Source Software has less features.	True
Open Source Software is less user friendly.	True

Table 2 – Features

On the frontend, in general proprietary software offers more features. This is as a result of focused product development and significant market share over many years. In some cases, this has been as a result of market lock-in as is the case with Windows and its associated applications, e.g. the Microsoft Office suite of applications. It may be easier to use peripherals with proprietary software. The Microsoft Office suite of applications has matured through years of added features, interface improvements, and usability studies.

Some open source programs are overtly influenced by proprietary products (e.g. Open Office is clearly influenced by Microsoft Office). These programs may be just as user friendly as the original. Of course, while most programs are flush with features, most users and organisations only use a handful of the features.

Having noted the lower feature capability and user friendliness, there are some significant exceptions to this general perception. An example of this is the Mozilla Firefox Browser. This browser has as many features or more than its main propriety competitor, i.e. Internet Explorer. In fact Internet Explorer is now playing catch-up in terms of its functionality. The higher

penetration of Internet Explorer is explained by the fact it is bundled with Windows in the consumer PC market. Also to an extent in the past Firefox has been seen as engineering/scientific based tools.

7.3 Capability

Open source is more reliable.	True
Open source is more secure.	True
Open source is more powerful.	Comparable with Propriety S/W
Open source is more network friendly.	True
Open source can be more customized.	True
Open formats are better.	True

Table 3 –Capability

Open source software may be more reliable than proprietary. It may not make as many errors or crash as often, e.g. Linux is famous for not crashing. The initial program may not be more reliable than a proprietary alternative, but it may mature faster as hundreds of programmers correct mistakes and add features. However, this advantage depends on the participation of enough competent programmers. Just like proprietary software, the reliability of an open source program depends on clear feedback after rigorous use in a variety of environments. Without enduring, sufficient and talented interest, an open source project may fail. In contrast, proprietary software companies may create and support necessary programs that no one would enjoy working on.

The transparency and reliability of open source may mean better security. With both open source and proprietary software, effective security depends on thoughtful deployment, regular monitoring, and timely upgrades or other modifications. Some proprietary solutions offer potentially robust security, but Linux leads the industry in defensive design. (This is a critical reason why business and government are interested in open source.) Transparency is sometimes misrepresented as a critical security threat. Transparency may make it easier to gain access to a program. However, the open source model leverages a community of programmers to maintain and improve security. This collective benefit seems to outweigh the danger of transparency. Also, the security of a system depends much more on careful deployment and maintenance.

Some open source programs are generally faster and scale larger than proprietary alternatives, especially where power means speed. Linux has a very strong capability to scale, i.e. to run large networks with many users. However, proprietary solutions will continue to offer

comparable speed and scalability. For example, Sun continue to develop proprietary technologies to push the speed and scaling envelope.

Since open source has been historically associated with the development of the Internet, much of its popularity of open source comes from its performance on the Internet. Open source software is often very network compatible. For example, more than half the World Wide Web runs on Apache web servers, an open source solution. In the main, open source networking solutions are compatible with proprietary software. For example, a lab of Linux computers can be seamlessly nested inside a larger proprietary network, using Samba an open source intermediary protocol.

By its very nature, open source allows any user or organization with enough expertise to tailor software to their needs. The diversity of Linux distributions reflects this flexibility. Each distribution offers a customized operating system targeting a specific market. There are even two competing graphical desktop interfaces: KDE and GNOME.

Open source usually uses open formats. As a result, open source programs are generally better at working together and sharing files. Open formats make integration much easier, in terms of file exchange, data exchange and application programming interface. The greatest potential advantage of proprietary formats is interface control. By controlling and licensing a format, a company should be able promote smooth file exchange between programs (e.g. the different programs in Microsoft Office). Open formats may mean a lower TCO, especially since the opportunity cost is high if an organisation is locked into a proprietary format. The open source movement is partly a response to incompatibility in proprietary software.

7.4 Deployment & maintenance

With open source you only pay for what you need.	True
Open source makes license management easier.	True
Open source means greater independence from companies.	True
Open source is harder to deploy.	Not proven
Proprietary offers better service & support.	True

Table 4 – Deployment & Maintenance

Open source software is usually more modular than proprietary alternatives. Modular software means each program is a discrete piece of a solution. If the user finds a better program, it can be swapped in without replacing the whole solution. Open source offers several interchangeable, modular programs for various needs. Users can find the best components and only deploy what they need. In contrast, proprietary solutions often come as packages, sometimes with features that users don't need. It can be difficult or impossible to replace a specific program for something smaller, more reliable, or otherwise more desirable.

License management is easier with open source. Users can install any number of copies, so open source companies don't bother with complicated licenses. There is no risk of illegal copies or license audits, and there are no anti-piracy measures (e.g. CD keys). Open source doesn't eliminate software management. Users should still track which version of what program is on which machine. The quality of any program, especially for security, depends on patches and other updates.

Open source users have more independence from software companies. Even if a software company goes bankrupt, the community still has the source code. This independence also means "end of life" decisions or undesirable new features can't be forced on the users. Since the software is open anyone can try to improve or support it. Users don't have to rely on a single vendor for all aspects of the solution. The same is true if a product moves in a direction incompatible with a user's needs, i.e. the user can choose to move to a different product.

Any software solution requires some expertise to deploy, secure, and maintain. In the past, open source software was perceived to be technically challenging and required considerable expertise. To an extent, the reason for this was that open source was focused on backend computing. Today, some open source software is just as easy to deploy as proprietary alternatives. For example OpenOffice is designed for entry- or mid-level user expertise, offering installers with graphical interfaces and suggested configurations.

Any software solution requires some service and support. For both open source and proprietary software, users depend on support sites as well as contracted support. The quality and availability of help is proportional to interest and use, especially in open source. The support costs for niche solutions are usually high. On the backend, open source is common so the community is large and helpful. A variety of companies offer help for using open source on the backend. On the frontend, open source is often still a niche. So it may be harder to find help and contracted help may be more costly.

A proprietary company may have a longer, better reputation for service and support. For example, it's relatively easy to find hardware drivers for Microsoft Windows solutions. Some open source companies offer good support contracts. It should be noted that both open source and proprietary support contracts provide limited accountability.

7.5 Users & Migration

Some open source software is just as easy to learn how to use.	True
Migration is expensive.	True
It's difficult to integrate open source & proprietary solutions.	True

Table 5 – Users & Migration

Software applications usage always includes a learning curve. People want computers to work like appliances, like washing Machines. Neither the proprietary nor the open source models

have produced solutions as reliable and user friendly as washing machines. For better or worse, most users are more familiar and more comfortable with proprietary software, especially Microsoft Windows due its wide spread use. Users may view open source solutions with uncertainty or fear. However, the situation is changing. For example, anyone familiar with Microsoft Office will probably find it easy to use OpenOffice for simple productivity. The latest versions of the Linux graphical user interface can create desktops with the look and feel of Microsoft Windows or Apple Macintosh. Also proprietary software may change from version to version, causing user discomfort, for example the move to Microsoft Office 2007 from previous versions, which has challenged even experienced users.

User discomfort is more prevalent among staff using frontend solutions. In contrast, backend users are generally more familiar with open source solutions or may adjust more readily. Frontend users can have very legitimate reasons to resist migration to open source. They may not have the time or energy to learn new software.

Migration from any existing solution to any new solution can be very expensive. The cost in monetary terms may be quantifiable, but the time and stress on staff can be uncertain. There may be unforeseen problems, as when existing hardware isn't actually sufficient or when data isn't easy to migrate (Don't forget/ignore this). While the price of open source software may be negligible, migration can still be costly. Users may believe that migration costs may be more than current costs, but the long-term savings are significant. This is especially plausible on the backend, where open source reduces or eliminates license fees.

Integrating any two programs is often challenging. It may be difficult to integrate some open source and proprietary solutions. This is usually because proprietary companies prefer their customers to use their brand for most or all of a solution. In many cases open source is a minority solution, so it's designed to integrate as smoothly as possible with proprietary solutions. However, integration may be too unreliable for practical use. For example, OpenOffice tries to use Microsoft Office file formats. Ideally, this allows an OpenOffice user to collaborate with Microsoft Office users. In practice, some files may not open cleanly in one program or the other. However, this problem also occurs when opening files from newer versions of Microsoft Office in older versions of the same program. A specific example of this it the fact that native Office 2007 file formats are not compatible with previous versions of office, a software adaptor for previous versions of Office is required to be deployed to achieve this. Added to this, it is now clear that Microsoft have begun their long road to use open file formats.

7.6 Free Markets & Choice

Software should be a commodity.	Not Proven
Open Source negates vendor lock in.	True
Open source negates monopolies.	True

Table 6 – Free Markets & Choice

Some open source proponents want software to be a commodity. Commodities are interchangeable resources without monolithic brand names or unique features. From the operating system up, most open source software is highly modular and "commoditized." Proponents criticize proprietary software for not being modular. Some companies try to monopolize a market by designing their software to be indispensable. This can stifle competitive service. If solutions are proprietary and interdependent, the only people who can offer service are those who learn the secrets and know all the programs. If users find a better component (e.g. a cheaper, faster, or more affordable program), they can't swap it in because the existing program isn't a commodity. Commodities are usually better for the consumer.

Ideally, if a company designs all aspects of a solution it should be the best possible solution. Quality and service should be associated with brand names but too often they aren't. When a market shifts to commodities consumers don't have to trust a brand name and may enjoy more competitive service. Companies and brands come and go but services exist as long as there is demand for them.

Lock in occurs when a user becomes dependent on a specific program from a specific vendor. Such a program is the opposite of a commodity. The program may not be modular enough to be replaced, or the user may have valuable data or other content stored in the program's proprietary format. When the program is indispensable or the data can't be migrated, the user is locked in to using that program forever. . Open source forecloses on this danger by using open formats and open code, i.e. lock in is nearly impossible with open source. For example, HTML is supposed to be an open format.

The competition among Web browsers illustrates the dangers of lock in on a much larger scale. Creating or hosting Web content with a proprietary solution sometimes restricts or denies access from another platform. For example, web page authors assert that Microsoft FrontPage is notorious for creating web pages that work well under the latest version of Internet Explorer, but not as well under other browsers. As the language of the Web, HTML is supposed to be an open format so that no single company can control the Web. If a proprietary program creates HTML skewed toward proprietary products (e.g. FrontPage and Explorer), better or less expensive software is excluded because it's less compatible and less reliable. Competition, innovation, and choice are stifled.

Open source proponents warn that proprietary software leads to monopolies through lock in and unfair business practices. Consolidation may suffocate innovation and lock in hurts choice. Virtual monopolies have less pressure to innovate and little pressure to lower prices. When it becomes a generic commodity, prices drop.

By comparison, the patents on pharmaceuticals have limited durations, so new proprietary drugs can eventually be replicated as generics. Consumers may depend on the drug, but when it becomes a generic commodity, prices drop. Consumers have more choice and may pay less. Some open source proponents want software to be like generic drugs. Proprietary software may have features or other value added, but the generic, open source software is a reasonable, thrifty choice.

7.7 Principles & Rights

The debate is really about philosophy, not money.	True
Software is better when it's transparent.	True
Open source is more empowering.	True
Open source supports intellectual property rights.	True

Table 7 – Principles & Rights

The original debate over free software was about philosophy. Proponents were more concerned about principles and rights than economics. However, the newer open source movement is more appealing to governments and companies. While they may find the philosophy attractive, they choose open source for very pragmatic reasons, i.e. costs and market share. The drivers are now about greater financial independence and potentially lower TCO. The debate is about money.

If software is transparent, any programmer can see what happens and why. Proprietary software is not transparent. Most computer users are familiar with the strange error messages that appear when a program fails. Often, these messages use special codes to express the problem. These codes point to underlying code without actually revealing what went wrong. Only someone who knows the underlying code can understand the codes. Because open source programs aren't trying to protect secrets, they can offer more exact information about an error. This allows any programmer to identify and fix problems. It's the suspicion of such errors that causes people to distrust proprietary programs.

Open source empowers the user more than proprietary software can. Any user can fix the bugs or add the features that matter most to them. A software company may decide a bug or feature isn't critically important to most of their users (or their sales). Open source frees the users to decide for themselves. This empowerment is most visible in backend solutions like servers and networks. The technicians who build and manage these solutions often have the freedom to choose any software they want as long as it works. These technicians are usually interested in the easiest, most powerful solutions. Many have chosen open source.

Opponents argue that open source threatens intellectual property rights. This is a misleading argument designed to scare people away from learning more about open source. It's inspired by the most radical open source license, i.e. the GPL. Source code released under the GPL can't be included in closed, proprietary software. But any content created using open source software still belongs to the author. For example, authors who create documents in OpenOffice still have copyrights on their work. With open source, only programmers can lose their intellectual property rights. They are willing to give up exclusive ownership of their code to contribute to the community and benefit from community-created programs.

Almost all software is created from existing software, using existing designs ("art") and common libraries of code. Hypothetically, if all software was licensed under the GPL then it would be almost impossible to create proprietary software without starting from scratch. But few people want or expect all software to become proprietary. The technology offers many potential advantages and challenges, but it's alarmist to equate open source with the theft of intellectual property.

8 The Future -

In the past there has been a perception that OSS is close to anarchy. This is not true, many established high profile companies are thriving by using and selling open source software. Open source represents a paradigm shift for the technology industry. OSS does and may undermine some existing businesses. There's little reason to buy a proprietary program when a comparable open source alternative exists. However, the software industry was already transforming from a product model to a service model. In the Telco space, Bull is a particular champion of open source, defining themselves as 'Architect of an Open World'TM. Bull creates open source components, develop existing open source components to deliver added value and act as an open source integrator. HP, through its Open Source Integrated Portfolio (OSIP) technologies, offers software, middleware, services, operating systems, and platforms. Red Hat sells open source software with value added, including service.

Open source is here to stay as it has reached a critical mass in its deployment. At this point, it is reasonable to assume that OSS will continue to permeate through the ICT industry, with its overpowering acquisition cost advantage, especially given the ongoing requirement to reduce product cost.

9 References

- 1) Open Source Initiative - <http://opensource.org>
- 2) The Open Source Definition - <http://opensource.org/docs/osd>
- 3) The Apache Software Foundation - <http://www.apache.org>
- 4) The Mozilla Foundation - <http://www.mozilla.org>
- 5) ITBusinessEdge - <http://www.itbusinessedge.com/topics/reader.aspx?oss=17361>
- 6) Standish Group International - <http://www.standishgroup.com>
- 7) Netcraft - http://news.netcraft.com/archives/web_server_survey.html
- 8) IDC – The Opportunity for Linux in a New Economy.
- 9) openEMS Overview - http://www.opennms.org/index.php/Main_Page
- 10) OpenOptions - http://www.netc.org/openoptions/pros_cons/index.html