THE MALAYSIAN GOVERNMENT INTEROPERABILITY FRAMEWORK for OPEN SOURCE SOFTWARE (MyGIFOSS)

MALAYSIAN PUBLIC SECTOR
OPEN SOURCE SOFTWARE (OSS)
INITIATIVE

FEBRUARY 2006

MAMPU

OSCC
OPEN SOURCE COMPETENCY CENTRE
MALAYSIA

MAMPU

MALAYSIAN PUBLIC SECTOR OPEN SOURCE SOFTWARE (OSS) INITIATIVE

The Malaysian Government Interoperability Framework for Open Source Software (MyGIFOSS)

COPYRIGHT

TABLE OF **CONTENTS**

PART A

PART B

PART C

LIST OF TABLES

1 INTRODUCTION

This document provides detailed recommendations resulting from the research and analysis conducted in the development of the Malaysian Government Interoperability Framework for Open Source Software (MyGIFOSS). It contains information on open source software (OSS), open standards and technical specifications recommended for adoption in Malaysia. MyGIFOSS is prepared as a supplement to the Malaysian Government Interoperability Framework version 1.0 (MyGIF), August 2003.

The intention of this document is to provide a guide for government agencies in the adoption of, and migration to, OSS within their ICT framework. It addresses issues concerning the use of OSS and proprietary software, integration of OSS with legacy systems, the exchange of information and data between heterogeneous systems, and compliance to published open standards where applicable. The emphasis for addressing the issues above is to ensure interoperability between systems and applications.

1.1 Definition

MyGIFOSS defines the minimum set of open standards and technical specifications governing the use of OSS and information access. In addition, it also contains application notes to guide the handling of migration issues regarding implementation of open source software in Government agencies.

MyGIFOSS covers the following aspects:

- Examples of OSS within the solution areas of the Malaysian Public Sector OSS Master Plan.
- Recommendations for Information and Services Access, covering standards availability.
- Application Notes – implementation guidelines that addresses the issues of using OSS and proprietary software within a heterogeneous environment.

Instead of creating new standards or specifications, MyGIFOSS adopts internationally recognised and ratified open standards, where available. For the purposes of this document, open standards are defined as standards which are international, transparent, unencumbered and published. These standards should also be recognised and ratified by bodies such as:

i. International Organization for Standardization (ISO)
ii. International Telecommunication Union (ITU)
iii. Institute of Electrical and Electronic Engineers (IEEE)
iv. The World Wide Web Consortium (W3C)
v. Internet Engineering Task Force (IETF)
vi. European Computer Manufacturers' Association (ECMA)
vii. Organisation for the Advancement of Structured Information Standards (OASIS)
viii. The Free Standards Group (FSG)

However, where no open standards are available, only well-published de-facto standards are applied.

1.2 Objectives

MyGIFOSS's objectives are:

- To enable proprietary and open source systems in different Government information systems, both within Government and external to Government, to communicate and inter-operate efficiently and effectively;
- To promote and foster the adoption of open source solutions within the Government, by emphasising the need for openness, transparency and competitiveness for all implementation of information systems;
- To promote and foster the adoption of open standards that enables the exchange of data between applications;
- To promote vendor-neutral and technology-neutral implementations, with the adoption of open standards, for all Government information systems; and
- To reduce the total cost of ownership of Government information systems, with the adoption of open standards.

1.3 Basis Of Recommendations

The key drivers guiding the recommendations of ICT standards and technical specifications for MyGIFOSS are:

- Interoperability:
  Standards and specifications recommended must be relevant to recommended use of OSS applications and the use of open standards for Information Access.
- Availability of internationally recognised standards:
  The standards, where available, must be recognised and adopted by internationally recognised bodies.

1.4 Scope

MyGIFOSS covers the use of OSS, standards for Information Access as well as application notes for guidance in usage and migration. The scope of the applications covered are within the six solution areas, as defined within the Malaysian Public Sector OSS Master Plan. The six solution areas are:

i. Workload Consolidation
ii. High Performance Computing
iii. Distributed Enterprise
iv. Application Solution

v. Infrastructure Solution

vi. Desktop Solution

MyGIFOSS standards and specifications should be considered for all new system implementations that fall within the solution areas as defined within the OSS Master Plan. For legacy systems that fall within the scope defined, agencies will need to assess if any integration are required between the legacy systems and other systems. If it is determined that integration would be required, interfaces will need to be defined to allow such integration to take place. The interfaces should take into consideration the guidelines contained within this document.

1.5 How To Use This Document

This document is structured in the following manner:

Part A: Chapters 2 and 3 provide technical specifications and standards for OSS.

Chapter 2: Aims to assist in the selection of OSS and contains examples for use within an agency's ICT implementation. A table detailing OSS examples are given, with their associated implementation and solution areas. Agencies are encouraged to refer to the Preferences and Guidelines column in the table when considering their implementations. The chapter then goes into greater detail on the software features, capabilities and shortcomings.

Chapter 3: Gives recommendations on information access governing the use of standards for data access and interchange. This chapter is relevant for agencies considering implementing software, either open source or proprietary, which requires information access and interchange. It gives recommendations to ensure availability of information and interoperability, allowing for different applications, systems and infrastructure to exchange information.

Part B: Chapters 4 and 5 are discussions on implementation issues and guidelines.

Chapter 4: The chapter discusses the considerations that need to be taken into account when planning for migration, without delving into the technical details.

Chapter 5: Discusses the use of OSS and proprietary software within a heterogeneous environment. It also provides technical guidelines for OSS implementation and migration.

Part C: Chapters 6, 7 and 8 conclude the document with:

Chapter 6: A brief conclusion of MyGIFOSS' objectives and purpose.

Chapter 7: A set of references used throughout the document.

Chapter 8: A list of abbreviations and acronyms used within the document.

PART A

2 OPEN SOURCE SOFTWARE WITHIN DEFINED IMPLEMENTATION AREAS

2.1 Overview

This chapter provides a selection of OSS to be used in an agency's ICT implementations. It includes a description on the features, rationale for inclusion and limitations of the software.

Within the Malaysian Public Sector OSS Master Plan, six solution areas were identified, as follows:

i. Workload Consolidation
ii. High Performance Computing
iii. Distributed Enterprise
iv. Application Solution
v. Infrastructure Solution
vi. Desktop Solution

The solution areas consist of individual software implementations, combining together to create a complete solution. These individual software implementations are grouped into the headings below, in which OSS examples are given:

■ **Operating System**

- Linux
- BSD

■ **Desktop Applications**

- Office Productivity Suite
- Project Management
- Mail Client
- Web Browser
- Multimedia Player
- Desktop Environment
- Educational Software

■ **Vertical Applications**

- Knowledge Management (KM)
- Content Management System (CMS)
- Enterprise Resource Planning (ERP)
- Document Management System (DMS)
- Hospital Information System (HIS)
- Land Information System (LIS)
- Workflow System

■ **Server Applications**

- Mail Transfer Agent (MTA)
- Mail Access Agent
- Groupware
- Web Server
- Remote Login Server
- Database Server
- Proxy Server
- File & Printer Server
- Backup Server / Tool

■ **Network Security**

- Firewall
- Network Intrusion Detection System (IDS)
- Virtual Private Network (VPN)
- Anti-virus
- Anti-spam

2.2 Guidelines/Standards Matrix

Table 2.1 below lists examples of OSS for use within identified implementation areas. The table is not meant to be exhaustive, but should represent a wide range of software under open source licenses. The subsections following the table go into greater detail with regards to the OSS examples given, providing information on the software, its rationale for selection, known limitations (if any) and implementation scenario examples.

No	Implementation Area	Solution Area Affected	OSS Software Example	Preferences and Guidelines
1	**Operating System**			
1.1	Linux	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution • Desktop Solution	• Red Hat • Fedora • Mandriva • Novell SuSE • Slackware • ELX • Debian • Gentoo	For Linux Distributions, preference should be made for those which follow the Linux Standard Base (LSB) 1.1, or higher, specifications.
1.2	BSD	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution • Desktop Solution	• FreeBSD • NetBSD • OpenBSD	There are three popular flavours of BSD, with the rest being niche offshoots. The user is advised to use the most relevant for their application.
2	**Desktop Applications**			
2.1	Office Productivity Suite	• Desktop Solution	• Open Office • KOffice • Abiword • GNUmeric	Data formats based on the Organisation for the Advancement of Structured Information Standards (OASIS) Open Document XML format should be used.
2.2	Project Management	• Desktop Solution	• MrProject • TaskJuggler • PHProjekt • TUTOS	Software chosen should be based on ability to provide required functions, e.g. GANTT charts, timekeeping, problem ticket tracking.
2.3	Mail Client	• Desktop Solution	• Evolution • Kmail • Thunderbird • Pine	Adherence to standard protocols, e.g. POP and IMAP, and other Internet RFCs are preferred.
2.4	Web Browser	• Desktop Solution	• Mozilla • Firefox • Konqueror • Epiphany	Adherence to W3C and related WWW protocols should be preferred.
2.5	Multimedia Player	• Desktop Solution	• MPlayer • Xine • xmms	Ability to play multiple formats is desirable. Should also be able to play open, patent-free media formats, e.g. Ogg Vorbis.
2.6	Desktop Environment	• Desktop Solution	• K Desktop Environment (KDE) • GNOME	The examples given here are the two most mature integrated desktop environment available for Linux and BSD. Use of one environment over the other depends on the user's preference.

No	Implementation Area	Solution Area Affected	OSS Software Example	Preferences and Guidelines
2.7	Graphic Manipulation	• Desktop Solution	• Gimp	The example given here is the most mature graphic manipulation tool available, works on many operating systems and support most of the graphic files format, e.g. gif, jpg, bmp, png.
2.8	Educational Software	• Application Solution • Desktop Solution	• Kalzium • KmPlot	Educational software are applications designed for teaching and learning. Suitability of the applications naturally depend on the requirements as determined by the educational system.
3	**Server Application**			
3.1	Mail Transfer Agent (MTA)	• Infrastructure Solution	• Sendmail • qmail • Postfix • Exim	The Mail Transfer Agents, generally referred to as mail servers, should conform to the Internet RFCs governing email over the Internet.
3.2	Mail Access Agent	• Infrastructure Solution	• UW IMAP • Courier IMAP	IMAP servers generally implement POP servers as well. The main purpose of IMAP/POP servers is to allow access to the mails stored in the mail servers.
3.3	Groupware	• Infrastructure Solution • Application Solution	• phpGroupWare • OpenGroupware	OSS Groupware vary in features and capabilities. Consider the best fit, and explore the possibility of having to make your own enhancements.
3.4	Web Server	• Infrastructure Solution	• Apache	Any web server must be able to serve pages which adhere to W3C specifications and guidelines.
3.5	Remote Login Server	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution • Desktop Solution	• OpenSSH	The example given here implements documented open standards for asymmetric and symmetric encryption. Any secure remote access software should implement the same.
3.6	Database Server	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution	• MySQL • PostgreSQL • Firebird	Relational Databases are the often at the core of many complex applications. Depending on the need, the choice of any particular database is largely dependent on the capabilities and requirements for the applications.
3.7	Proxy Server	• Infrastructure Solution	• Squid	Proxy servers caches web and ftp traffic. They help to reduce bandwidth requirements. The use of proxy servers are recommended.
3.8	File & Print Server	• Infrastructure Solution	• SAMBA • Linux NFS • BSD NFS • Common UNIX Printing • System (CUPS)	The Network File System (NFS) and CUPS are standard on all UNIX and UNIX-like operating systems, thus enabling interoperability between these systems. Some non-UNIX proprietary operating systems do not implement NFS or CUPS, but use SMB/CIFS instead. Use of SAMBA is recommended where interoperability with these systems are required.

No	Implementation Area	Solution Area Affected	OSS Software Example	Preferences and Guidelines
3.9	Backup Server / Tool	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution • Desktop Solution	• Amanda	Backup software has to have capability to do incremental backups and network spanning across servers or storage area networks. Further, it must have the ability to handle various operating systems and backup devices, such as disks, tapes and optical libraries and multichangers. Analysis of the level of backups are needed before obtaining any particular software.
4	**Network Security**			
4.1	Firewall	• Infrastructure Solution	• Linux iptables • FreeBSD ipfilter • IPFW • OpenBSD packetfilter	Firewalls are generally regarded as the first line of defense for Internet security. There are two kinds of firewalls – software and hardware. Software firewalls are implemented within general purpose servers while hardware firewalls are special purpose appliances. The choice of a firewall implementation is often down to “high-level” considerations, like ease of configuration and maintenance since the capabilities are often similar across platforms.
4.2	Network Intrusion Detection System (IDS)	• Infrastructure Solution	• Snort	Network IDS often acts as a forensic tool for analysing network breaches. It requires active monitoring as well as constant updating of signatures to be effective. When considering a Network IDS, those which are being actively updated with intrusion signatures are recommended.
4.3	Virtual Private Network (VPN)	• Infrastructure Solution	• OpenVPN • FreeS/WAN	VPNs are used as a cheaper alternative to having costly lease-lines to connect distance branch offices and sites. The objective of a VPN is to leverage on the low-cost of having general Internet connections while having the benefits of security and privacy for internal traffic. A VPN implementation must have strong encryption and mechanisms for trust and authentication.
4.4	Anti-virus	• Infrastructure Solution • Desktop Solution	• ClamAV	OSS operating systems and applications are not generally under threat of viruses, worms and trojans. However, they are often used as gateways and servers for proprietary operating systems and applications which are vulnerable. As such, anti-virus scanners for these proprietary software have been created and deployed on OSS systems. Anti-virus software which have are actively developed and have the virus signatures constantly updated are highly recommended.

No	Implementation Area	Solution Area Affected	OSS Software Example	Preferences and Guidelines
4.5	Anti-spam	• Infrastructure Solution • Desktop Solution	• SpamAssassin • Built-in spam filters within Evolution, Thunderbird and Kmail	Spam filtering are traditionally executed at the mail server, however, there is now an increasing trend to include spam filtering capabilities on the mail clients themselves. OSS mail clients often have extensive filtering capabilities. When considering a mail client, those with spam filtering capabilities are recommended for installation.
5	**Vertical Application**			
5.1	Knowledge Management (KM)	• Application Solution	• phpwiki • zwiki • mediawiki	KM systems are an attempt to capture and retain knowledge within an organisation. It is most successful in "flattening" an organisation's hierarchy, giving equal opportunities for everyone within to contribute and disseminate information and knowledge. When choosing a KM system, the user is advised to consider the level of access control and moderation needed since the philosophy and implementation of KM systems vary widely in terms of the eligibility of users to input and edit information.
5.2	Content Management System (CMS)	• Application Solution	• Phpnuke • mambo • typo3 • opencms	CMSes are deployed as easy-to-use applications for websites and portals. There are many CMSes on the market today and the user is advised to carefully consider the features and capabilities of each one. CMSes which produce open standards compliant web pages are recommended.
5.3	Enterprise Resource Planning (ERP)	• Workload Consolidation • High Performance Computing • Distributed Enterprise • Application Solution • Infrastructure Solution • Desktop Solution	• Fisterra • Compiere	Enterprise Resource Planning (ERP) applications are typically large and requires a lot of customisation to suit any particular enterprise. It is often difficult to determine which of the ERP subsets that one might need. This is not helped by the fragmentation of ERP itself into, for e.g. Customer Relations Management, Supply Chain Management, Manufacturing Resource Planning, Partner Relations Management and other higher level subsets like business intelligence and KPI dashboards. When considering an ERP system, careful planning and attention to process organisation are needed.

No	Implementation Area	Solution Area Affected	OSS Software Example	Preferences and Guidelines
5.4	Document Management System (DMS)	• Application Solutions	• KnowledgeTree • Eidetic • Terracotta	Document Management Systems help in organising and archiving official documents and other related collateral, such as presentations, drawings and designs. The use of a DMS is needed for efficient and effective management of an organisation's intellectual property.
5.5	Hospital Information System (HIS)	• Application Solution	• OpenVISTA • Care2x	Hospitals are complex organisations with multiple levels of operations – usually divided into clinical and non-clinical operations. The administration of a hospital benefits greatly with the implementation of an HIS. When choosing an HIS, it is important to clearly have well defined SOPs and to evaluate whether the HIS'es considered have the required flexibility and scalability to implemented the SOPs.
5.6	Land Information System (LIS	• Application Solution	• osrs • grass	Comprehensive Land Information Systems (LIS) are still in infancy but there are individual applications which handle different aspects needed for land and real estate management. Among the more mature ones are Geographical Information Systems (GIS). Choosing an LIS would involve similar processes to choosing an HIS above, with great importance on having well defined SOPs.
5.7	Workflow System	• Application Solution	• Bonita	Bonita is a flexible cooperative workflow system, compliant to WfMC specifications, based on the workflow model proposed by the ECOO Team, which incorporates the anticipation of activities as a more flexible mechanism of workflow execution. Bonita is Open Source and is licensed under the LGPL.

Table 2.1: Implementation Guideline Matrix

2.3 Operating Systems

2.3.1 LINUX

Background description:

Linux consists of the Linux kernel (core operating system - OS), originally written by Linus Torvalds, along with utility programs developed by the Free Software Foundation and others. The combination of thousands of OSS and the Linux kernel makes it possible for a functional operating system to be created. Since then, many commercial and community driven projects were spawned to create new operating systems based on the Linux kernel and associated OSS. The differences in how they were created, the methodology taken to create them and the philosophy behind the operating systems created, are now known as Linux distributions. In essence, the three different "uses" of the word Linux can be summarised as:

i. Kernel – the kernel is the central program of any operating system. This is the program that talks directly to the hardware of the computer. Other programs make requests of the kernel to get something done on the computer. For instance, in order to put a character on the computer screen, a program must ask the kernel to put a certain character in a certain place on the display screen. (There are exceptions to this, but in general, all communication to the computer hardware goes through the kernel.)

ii. Operating System - the combination of the kernel and the application software used to interact with the kernel.

iii. Distribution – the many different flavours of Operating Systems created by thousands of other software packaged together with the Linux kernel.

Further information on the Linux kernel, operating system and distributions can be found at the Linux Documentation Project website: http://www.tldp.org/

The specific document that answers the frequently asked questions on Linux can be found at:
http://tldp.org/FAQ/Linux-FAQ/index.html

Table 2.2 on page 14 gives a non-exhaustive comparison of available Linux distributions, giving among other things their compliance to the Linux Standard Base (LSB) and computer architectures supported.

A note on the Linux Standard Base:

From the Linux Standard Base web site (http://www.linuxbase.org/):

Mission Statement

To develop and promote a set of standards that will increase compatibility among Linux distributions and enable software applications to run on any compliant system. In addition, the LSB will help coordinate efforts to recruit software vendors to port and write products for Linux.

What the statement essentially means in the practical sense is that the LSB seeks to remove the uncertainty in choosing a Linux distribution. Distributions which are LSB compliant retains the ability to run any applications which are developed following the LSB specifications.

As such, when choosing a Linux distribution for implementation, it is strongly recommended that a LSB compliant distribution is chosen.

Distribution	LSB Compliance	Vendor/ Community Support	Multimedia Support	Package Management	Auto Update	Arch Support
Red Hat	1.3	Vendor	Limited	Rpm	Yes	i386, ia64, ppc, s390, s390x, x86-64
Fedora	No	Community	Limited	Rpm	Yes	i386, x86-64
Novell SuSe	1.3, 2.0	Vendor	Yes	Rpm	Yes	X86, x86, PowerPC, m68k, ARM, MIPS
Mandrake	No	Vendor/Community	Yes	Rpm	Yes	i586
ELX	No	Vendor	Yes	apt-get (synaptic)	Yes	X86-486
Slackware	No	Community	Yes	tgz	Yes	i486
Debian	No	Community	Yes	apt-get, dpkg	Yes	Alpha, Arm, HPPA, i386, IA64, m68k, Mips, PPC, S390, Sparc
Gentoo	No	Community	Yes	emerge	Yes	x86, amd64, hppa, ppc, ppc64, sparc

Table 2.2: Comparison of Available Linux Distributions

Red Hat/ Fedora	
Description	• Both are developed and maintained by Red Hat, Inc. • Red Hat Linux is the commercially supported version by Red Hat Inc. • Fedora Linux is the community version, which are distributed without any commercial support from Red Hat Inc.
Reference	• http://www.redhat.com/
Rationale for selection	• They are most well-known. • They are easy to install. • There are excellent community supports. • There are commercial supports.
Limitations	• Poor multimedia support due to Red Hat Inc.'s decision not to include patented software and algorithms, e.g. MP3, Windows Media formats, Apple's Quicktime.
Implementation Scenario Example	• Red Hat Linux distributions are commonly used in server environments. This is due to its large support for commercial based software like Oracle and IBM WebSphere.

Mandriva	
Description	• Formerly known as Mandrake, it was created in 1998 by Gael Duval, with the goal of making Linux easier to use for everyone. • Mandriva is the result of Mandrake acquiring Conectiva of Brazil, another Linux distribution company.
Reference	• http://www.mandriva.com/
Rationale for selection	• Mandriva is easier to use for non-technical desktop users. • There are enormous community supports. • It is user-friendly with graphical configuration utilities.
Limitations	• It is limited to x86 architectures.
Implementation Scenario Example	• Mandriva Linux Distribution is commonly used on desktop environments. This is mainly due to its robustness, attractive graphics and user friendliness.

SuSE	
Description	• Distributed by a German company, SuSE Linux AG. Very popular in Europe. • It is one of the fastest growing distributions worldwide. • It was recently purchased by Novell Inc.
Reference	• http://www.suse.com/
Rationale for selection	• SuSE is widely implemented in European governments and corporations. • It supports multiple architectures, e.g. x86, PowerPC, IBM Mainframes. • It is easy to install. • It has special desktop edition, i.e. Novell Linux Desktop.
Limitations	• No known major limitations exist.
Implementation Scenario Example	• SuSE Linux distributions are mostly used because of its comprehensive system and network administration tools. It has excellent commercial support on various platforms.

Slackware	
Description	• One of the earliest Linux distributions, started by Patrick Volkerding in 1993.
Reference	• http://www.slackware.com/
Rationale for selection	• Slackware is favoured by many system administrators for servers, due to its minimalist design. • It is one of the older Linux distributions, proving its stability. • It does not suffer from the package dependency issues of RPM-based distributions.
Limitations	• Minimalist design and simple installation tools means that there is a steeper learning curve. • Its support options not as comprehensive as Red Hat, SuSE or Mandriva. • It supports only x86 hardware.
Implementation Scenario Example	• Slackware are mainly used on servers which require minimal user interaction. It is often the distribution of choice for edge servers, i.e. firewalls, intrusion detection systems etc.

2.3.2 BSD

Background description:

BSD stands for “Berkeley Software Distribution”, the UNIX derivative distributed by the University of California, Berkeley beginning from the 1970s. It is also used collectively to describe the modern descendants of the original distribution. The BSD family of operating systems provides a number of complete operating systems packages, the three most popular being FreeBSD, NetBSD and OpenBSD.

FreeBSD	
Description	• An advanced OS derived from BSD, the version of UNIX developed at the University of California, Berkeley. • It is managed by the FreeBSD Foundation.
Reference	• http://www.FreeBSD.org/
Rationale for selection	• FreeBSD is ideal for internet and intranet servers. • It has features of high performance and ease of use by end users. • It is optimized for x86 platform. • It has a lot of Linux applications ported. • It has the ability to run Linux binaries.
Limitations	• No known limitations exist.
Implementation Scenario Example	• FreeBSD is used as an alternative to Linux operating Systems. • It is commonly used where the users are more comfortable with native UNIX features. • Due to the development methodology, FreeBSD is not likely to fragment and fork.

NetBSD	
Description	• NetBSD is a free, secure, and highly portable UNIX-like Open Source operating system available for many platforms. Its OS is also derived from *BSD, the version of UNIX developed at the University of California, Berkeley. • It is managed by the NetBSD Foundation.
Reference	• http://www.netbsd.org
Rationale for selection	• NetBSD has a wide range of support for many platforms. NetBSD is designed to take advantage of the latest high end hardware available in Alpha, PowerPC, and PC systems, while still retaining support for older architectures. • The entire kernel and the core of the userland utilities are shipped under a BSD license. This allows companies to develop products based on NetBSD without the requirement to make the changes public.
Limitations	• No known limitations exist.
Implementation Scenario Example	• NetBSD is used as an alternative to Linux Operating Systems. • It is commonly used where the users are more comfortable with native UNIX features. • NetBSD has a wide range of support for various platform makes it suitable for research and development environments.

OpenBSD	
Description	• OpenBSD is a free, multi-platform 4.4BSD-based UNIX-like Operating System. It emphasizes portability, correctness, and integrated cryptography.
Reference	• http://www.openbsd.org/
Rationale for selection	• OpenBSD is the only Operating System confident enough to claim that it has only one remote hole in its default install for the last 8 years. • It is exported with cryptography, making it suitable for developing cryptography applications. • Its Operating System includes integrated cryptography applications for software and hardware.
Limitations	• No known limitations exist.
Implementation Scenario Example	• OpenBSD is used as an alternative to Linux Operating System. • It is often used as firewalls, since its default install is considered to be secure. • It is exported with cryptography, making it suitable for developing cryptography applications.

2.4 Desktop Applications

2.4.1 Office Productivity Suite

Background description:

An Office Suite is a group of applications, consisting of a text editor, spreadsheet, presentation slides creator and optionally a simple database that are used as productivity tools

OpenOffice

Description	• OpenOffice originated from a proprietary office suite, StarOffice. • StarOffice was a product from a company called StarDivision GMBH, which was acquired by Sun Microsystems in 1999. • Sun Microsystems then open sourced StarOffice, and renamed it OpenOffice.org. • There are still successive versions of the proprietary StarOffice, using the same code base as OpenOffice. • Differences between the two can be found at: http://www.openoffice.org/FAQs/mostfaqs.html#7
Reference	• http://www.openoffice.org/
Rationale for selection	• OpenOffice has a similar user interface with popular proprietary office suites. • It has features comparable to Microsoft Office. • It can read and edit Microsoft Office files format. • Supports the OASIS OpenDocument format.
Limitations	• Reading and writing to Microsoft Office file formats by using OpenOffice may not be reproduced perfectly. • OpenOffice may have limited fonts and graphical images on platforms other than Microsoft Windows.
Implementation Scenario Example	• OpenOffice can be installed on Linux, Solaris, FreeBSD or Windows Operating Systems. • It allows co-existence with Microsoft Office Suite when it is being installed on Microsoft Windows. • In an office environment, it can be used for all desktop computers.

2.4.2 Project Management

Background description:

Project Management software aims to assist the management of projects by providing tools to plan and monitor timelines, milestones, resources and budget, among others.

MrProject

Description	• MrProject is developed by Code Factory. • A tool using Gantt charts to visualize project timelines, milestones and progression. • Some other features are task dependencies, resource tables, allocation and groups.
Reference	• http://mrproject.codefactory.se/
Rationale for selection	• It is the closest alternative tool to Microsoft Project.
Limitations	• Unable to open other project management programs' file formats.
Implementation Scenario Example	• MrProject can be installed for project management purposes. • In a coordinated project environment, MrProject can help organize and manage people, resources and time. To use its full features, PostgreSQL database needs to be installed along with Linux clients.

2.4.3 Mail Client

Background description:

Mail clients are formally referred to as Mail User Agents (MUA). Their purpose are for retrieving, reading, composing and replying/sending emails.

Evolution	
Description	• Evolution is developed by Ximian to provide a unified email client, similar to Microsoft's Outlook • Ximian was recently acquired by Novell Inc.
Reference	• http://www.ximian.org/
Rationale for selection	• It is the closest alternative tool to Microsoft Outlook. • It comes with Personal Information Manager (PIM) feature. • It supports GNU Privacy Guard. • It supports Microsoft exchange server with the Ximian connector.
Limitations	• No known limitations.
Implementation Scenario Example	• Evolution can be installed on Linux or FreeBSD Operating System. • In an office environment, it can be used on all desktop computers running an OSS or UNIX operating system.

2.4.4 Web Browser

Background description:

A program to surf the Internet.

Mozilla and Firefox	
Description	• Mozilla Firefox is a web browser that originated from Netscape Navigator, which was open sourced by the Netscape corporation. • The Navigator code was considered too complicated and a complete rewrite was made, resulting in Mozilla. • Mozilla is the base for several other derived web browsers, e.g. Firefox, Netscape, Galeon. • Mozilla comes with email client, chat tools, html composer and news reader. • Firefox is the browser only.
Reference	• http://www.mozilla.org/
Rationale for selection	• Firefox is now the most popular OSS browser. • It is the most standard compliant OSS web browsers. • It has extra functionalities which include high privacy and security encryption, pop ups and junk mail stopper, and tabbed browsing.
Limitations	• Inability to access some Internet Explorer specific websites, especially those containing ActiveX.
Implementation Scenario Example	• Firefox can be installed on Linux, FreeBSD or Windows Operating Systems. • It also allows co-existence with Internet Explorer as well as other commercial browsers when it is being installed on Microsoft Windows. • In an office environment, it can be used for all desktop computers.

Konqueror	
Description	• Konqueror was developed by the KDE group to be their default file and web browser. • It uses its own rendering engine, KHTML, but can also use Mozilla's Gecko to render web pages. • The rendering engine is constantly improved and is now used by Apple Computers Inc. for their Safari browsers.
Reference	• http://www.kde.org/
Rationale for selection	• Konqueror integrates local and external browsing. • Its rendering engine is under constant enhancements. • It has the ability to change browser identification, depending on website visited. Therefore, Konqueror has the ability to present itself as Internet Explorer, Mozilla or any other browser. • It has several other innovations, such as integrated translation tools, tabbed browsing etc.
Limitations	• Konqueror may not correctly render Internet Explorer specific pages.
Implementation Scenario Example	• Konqueror can be installed on Linux or FreeBSD. • In an office environment, it can be used on desktop computers that uses any OSS or UNIX operating systems

2.4.5 Multimedia Player

Background description:

A multimedia player for audio and video

MPlayer	
Description	• MPlayer is developed to be a player that can play any type of known video formats using pre-existing codecs
Reference	• http://www.mplayerhq.hu/
Rationale for selection	• For i386 architecture-based machines, MPlayer can utilise codecs used by Windows-based media players. • MPlayer's default supported formats are MPEG, AVI, VCD and DVD. • It has extra supported formats (codecs required) for Apple Quicktime Movie (MOV), RealPlayer (RM), Microsoft Windows Media (WMV) and others.
Limitations	• When using certain codecs, MPlayer may not be distributable under an open source license.
Implementation Scenario Example	• MPlayer can be installed on any OSS or UNIX operating system. • With the right decoder files installed, it can view almost any file formats including MOV, RM and WMV.

xine	
Description	• xine is an open source mpg and vcd player.
Reference	• http://Xinehq.de/
Rationale for selection	• It has a user-friendly interface. • It has a wide range of supporting format.
Limitations	• It cannot play certain video formats, which do not have Linux or FreeBSD codecs. • It cannot play audio-only files.
Implementation Scenario Example	• xine can be installed on any OSS or UNIX operating system.

xmms

Description	• xmms is an open source audio player, which has a similar interface to WinAmp.
Reference	• http://www.xmms.org/
Rationale for selection	• It supports MP3, Ogg Vorbis, WAV and audio CD formats.
Limitations	• To support MP3, xmms requires a certain library. Since MP3 is a patented product, some Linux distributions do not include it.
Implementation Scenario Example	• xmms can be installed on any OSS or UNIX operating system. • It can play many different audio file formats including WAV, MP3 and Ogg Vorbis.

2.4.6 Desktop Environment

Background description:

A desktop environment is a graphical user interface (GUI) system which sits on top of Linux, BSD or other UNIX-like operating systems. For these OSes, the GUI is not tightly coupled with the kernel, as in Microsoft's Windows environment. As such, there are several desktop environments that can be used.

KDE

Description	• KDE or K Desktop Environment is an open source project initiated by Matthias Ettrich in 1996. • It was created to provide a uniform, consistent and user-friendly desktop environment for UNIX and UNIX-like operating systems.
Reference	• http://www.kde.org/
Rationale for selection	• KDE has a "click, drag and drop" capability. • Its administration is easy with its bundled programs. • It supports other open source environment software.
Limitations	• No known limitations exist.
Implementation Scenario Example	• KDE can be installed as the desktop environment of choice for OSS and UNIX operating systems.

GNOME

Description	• After the KDE project was started, some Free Software advocates raised the issue of licensing. Even though KDE was licensed under the GPL, its core library, QT (http://www.trolltech.com/) was not. Furthermore, the OSS license for QT was considered incompatible with the GPL. • A parallel desktop environment, developed with GTK+, a GPL'ed library, was initiated. The project was, and still is, led by Miguel de Icaza. • The project is called GNOME, short for the GNU Network Object Model Environment. • It was also created to provide a uniform, consistent and user-friendly desktop environment for UNIX and UNIX-like operating systems.
Reference	• http://www.gnome.org/
Rationale for selection	• It has a "Click, drag and drop" capability. • Its administration is easy with its bundled programs. • It supports other open source environment Software.
Limitations	• No known limitations exist.
Implementation Scenario Example	• GNOME can be installed as the desktop environment of choice for OSS and UNIX operating systems.

2.4.7 Graphic Manipulation

Background description:

Software that can be used to perform various types of graphical manipulation.

Gimp	
Description	• Gimp is the most mature OSS graphic manipulation tool • Works on many operating systems, including Linux, BSD, Windows and Mac OS X.
Reference	• http://www.gimp.org
Rationale for selection	• It supports most of the graphic files format, e.g. gif, jpg, bmp, png.
Limitations	• No known limitations exist.
Implementation Scenario Example	• It is used for creating graphical images and editing graphics. • It can be used as a replacement for proprietary image manipulation software.

2.4.8 Education Software

Background description:

Software used for educational purposes for various fields such as chemistry, mathematics, physics.

Kalzium	
Description	• Kalzium is an information database which shows the periodic system of elements for chemistry studies.
Reference	• http://www.kde.org
Rationale for selection	• It can be used to visualize the Periodic Table of the Elements by blocks, groups, acidic behavior or different states of matter. • It can also be used to plot data for a range of elements (weight, mean weight, density, IE1, IE2, electronegativity). • It can go back in time and see what elements were known at a given date.
Limitations	• No known limitations exist.
Implementation Scenario Example	• It is used in chemistry classes.

KmPlot	
Description	• It is a mathematical function plotter.
Reference	• http://www.kde.org
Rationale for selection	• KmPlot is a mathematical function plotter for the KDE Desktop. It has a powerful built-in parser. It can plot different functions simultaneously and combine them to build new functions. • It supports parametric functions and functions in polar coordinates. Several grid modes are supported. Plots may be printed with high precision in the correct scale.
Limitations	• No known limitations exist.
Implementation Scenario Example	• It can be used for mathematical applications.

2.5 Server Applications

2.5.1 Mail Transfer Agent (MTA)

Background description:

Email servers are formally referred to as Mail Transfer Agents (MTA). Their purpose are to receive and distribute emails to their correct destinations. An email server can also serve as a relay, i.e. it forwards emails between locations when direct routing is not possible.

Sendmail	
Description	• Sendmail is written by Eric Allman while a researcher at the University of California at Berkeley. • It is the most popular UNIX-based MTA.
Reference	• http://www.sendmail.org
Rationale for selection	• It is included in most OSS operating systems. • It has the ability to disable open relaying. • It has spam filtering capabilities.
Limitations	• No known limitations exist.
Implementation Scenario Example	• Sendmail can be used to replace Microsoft Exchange server for mail transport. • However, there are some migration issues that might effect Microsoft Exchange groupware functionality.

qmail	
Description	• qmail is written by Dan Bernstein while a researcher at the University of Illinois, Chicago. • It is the second most popular UNIX-based MTA
Reference	• http://www.qmail.org
Rationale for selection	• It is included in most OSS operating systems. • It has good mailing list management capabilities. • It claims to be more secure that Sendmail. • It is very efficient and fast.
Limitations	• No known limitations exist.
Implementation Scenario Example	• Qmail can be used as an alternative to Sendmail.

Postfix	
Description	• Postfix is written by Wietse Zweitze Venema while working at IBM's Thomas J.Watson Research Center, USA. • It is another alternative UNIX-based MTA
Reference	• http://www.postfix.org
Rationale for selection	• It is included in most OSS operating systems. • It has the ability to disable open relaying. • It has spam filtering capabilities. • It has a similar configuration scheme to the Apache web server.
Limitations	• No known limitations exist.
Implementation Scenario Example	• Postfix can also replace Microsoft Exchange server for mail transport. • However, there are some migration issues that might effect Microsoft Exchange groupware functionality.

2.5.2 Mail Access Agent

Background description:

Mail Access Servers are software which implements protocols that allow mail clients to retrieve emails from a mail server. The most used protocols are Post Office Protocol (POP) and Internet Message Access Protocol (IMAP). IMAP is the more advanced protocol. It supports online and disconnected/ offline access. POP only supports offline access.

UW IMAP	
Description	• UW IMAP is Developed by University of Washington to implement the IMAP protocol.
Reference	• http://www.washington.edu/imap/
Rationale for selection	• UW IMAP has a full featured imap server. • It also supports POP. • It is the default imap server for most Linux distributions.
Limitations	• No known limitations exist.
Implementation Scenario Example	• IMAP is mainly used as a mechanism for retrieving electronic mail from the servers. • It is most suitable for use by users accessing emails from multiple clients and machines.

2.5.3 Groupware

Background description:

Software that enables a group of users to collaborate on a project by means of sharing resources.

phpGroupWare	
Description	• phpGroupWare is a multi-user web-based groupware suite written in PHP.
Reference	• http://www.phpgroupware.org
Rationale for selection	• phpGroupWare has production (stable) version provides an API for developing additional applications. • It has a built-in installation program. • It has tight security controls via Access Control List (ACL). • It has multi-language support using gettext. • It has a Preferences System which has the ability to allow users to change preferences only for current session (does not save to database).
Limitations	• phpGroupWare was not designed to handle groups for electronic commerce transactions.
Implementation Scenario Example	• phpGroupWare requires a database server to manage its data. • It also requires a Web Server to be installed with PHP support in it. • In most cases, phpGroupWare can be implemented to handle community, office or departmental groups.

2.5.4 Web Server

Background description:

An application to host website content. Some web servers can be integrated with other middle tier applications or tools to provides web-based applications.

Apache	
Description	• Apache is an HTTP or Web Server. • It can run on UNIX and Microsoft Windows platform. • Apache started when Brian Behlendorf started collecting patches to be applied to the last version of NCSA in 1995. • Hence the name Apache came from "A patchy server" (this is an apocryphal anecdote). • The Apache Software Foundation was formed in June of 1999 to maintain the Apache project. • Now Apache is known as a robust, commercial-grade, featureful, and freely-available source code implementation of an HTTP (Web) server.
Reference	• http://www.apache.org/
Rationale for selection	• Currently, 67% of the web servers worldwide us Apache. • It supports development tools such as php, cgi (perl, c++). • It is modularly designed, i.e. modules can be added to have extra functionality.
Limitations	• No known limitations exist.
Implementation Scenario Example	• Apache Web Server is very robust web server. • It has various extensions that can support many different languages. • In an Information Technology infrastructure that does not use proprietary or Microsoft ASP or .NET technology, it can be used to replace any other web server. • On Information Technology infrastructure that uses Microsoft ASP or .NET technology, migration issues can occur.

2.5.5 Remote Login Server

Background description:

Remote login is a method to access other machines. This is usually done for convenience of administration and maintenance. An unencrypted login can compromise the remote machine security. Therefore, use of secure methods are recommended.

OpenSSH	
Description	• OpenSSH is the open source tool implementing the ssh protocol. • The ssh protocol is a secure alternative to telnet, rlogin and ftp. OpenSSH was originally written for OpenBSD.
Reference	• http://www.openssh.org/
Rationale for selection	• OpenSSH's transactions are encrypted. • It is the default remote login used by all Linux distributions and BSD flavours. • It comes with extra tools such as scp for remote copy and sftp for implementing file transfer protocol (ftp) securely. • It supports other open source environment software.
Limitations	• No known limitations exist.
Implementation Scenario Example	• OpenSSH is arguably the most secure way to access a remote machine. • It supports most operating systems, and has a variety of clients for different kinds of operating systems. • It allows secure login and file transfer from one host to the other.

2.5.6 Database Server

Background description:

A database is an organised collection of data, and a database engine as applies to software is a system for the organisation and collection of data. Databases form an important component within an organisation, ranging from ERP, CRM, to web-based applications and portals.

MySQL

Description	• MySQL is an open source relational database. It is develop by MySQL AB, a company from Finland. • MySQL comes with two licenses, a GPL-ed version and a proprietary one. • For applications that require support and the ability to make modifications to the source code without releasing the changes, the proprietary license can be used.
Reference	• http://www.mysql.org/
Rationale for selection	• MySQL is lightweight and less resource hungry. • It is the most popular database for PHP-based applications. • It works with many other development tools. • Third-party tools are available to provide a graphical user interface for administration. • It can be installed on Windows as well as Linux/FreeBSD Operating System.
Limitations	• No known limitations exist.
Implementation Scenario Example	• MySQL can be used as a replacement for proprietary databases in just about any scenario. • In large-scale implementations, hardware requirements should be given due consideration. • In the case of migrating from other databases to MySQL, a few migration issues might cause some problems, especially where non-ANSI SQL calls are frequently used.

PostgreSQL

Description	• PostgreSQL is a free RDBMS, originally started in 1985 by Micheal Stonebreaker at University of California, Berkeley. • It was originally initiated as a more modern and contemporary version of another database called Ingres. • It is released under a flexible BSD-style license.
Reference	• http://www.postgresql.org/
Rationale for selection	• PostgreSQL is a full featured DBMS database. • It is able to handle large volumes of data. • It works with many development tools.
Limitations	• No known limitations exist.
Implementation Scenario Example	• PostgreSQL can be used as a replacement for proprietary databases in just about any scenario. • In large-scale implementations, hardware requirements should be given due consideration. • In the case of migrating from other databases to PostgreSQL, a few migration issues might cause some problems, especially where non-ANSI SQL calls are frequently used.

2.5.7 Proxy Server

Background description:

Proxy servers act as a form of protection for client machines accessing untrusted web and ftp servers. It achieves this by acting as the intermediary between a client requesting information and the server providing the information, in the process masking the client's identity.

As an added benefit, a proxy server can also act as a cache server, storing the information requested by a client, such that when it, or other clients, request the same information, the proxy then serves the information from its own cache without having to access the remote server. This saves on bandwidth.

Squid	
Description	• Squid is a full-featured open source proxy and cache server designed to run on UNIX systems. • It can also be installed on Microsoft Windows.
Reference	• http://www.squid-cache.org/
Rationale for selection	• It supports proxying and caching of URIs (e.g. HTTP, FTP) • It supports proxying for SSL cache hierarchies ICP, HTCP, CARP, Cache Digests. • It supports transparent caching and HTTP server acceleration. • It supports extensive access controls.
Limitations	• No known limitations exist.
Implementation Scenario Example	• Squid is a proxy server to cache heavily accessed files on the network. • It works well with other applications like Apache Web Server to cache web pages. • A web server runs together with a squid server to minimize the load on the web server while data is being accessed from the hard disk. This will provide better performance on the hosted web pages and also lengthen the life of the hard disk.

2.5.8 Files & Print Server

Background description:

File & print servers allow the sharing of resources within a networked environment. A file server consolidates data storage into a centrally managed storage system, allowing easier maintenance, backup and upgrading. A print server allows multiple clients to share printers. Further, print servers can additionally have the capabilities to allow and restrict access to printers, and keep an audit of printer usage.

SAMBA	
Description	• SAMBA is a file and print server for Windows clients using SMB or CIFS. • It is the link between Windows and Linux machines. With SAMBA, files and printers can be shared on these two different OS. • It is being actively developed by a global team of about 30 active programmers and was originally developed by Andrew Tridgell.
Reference	• http://www.SAMBA.org/
Rationale for selection	• SAMBA is the most widely used software to enable printer and file sharing between UNIX and Microsoft Windows environments. • It can act as a PDC (Primary Domain Controller) for Microsoft Windows machines.
Limitations	• No known limitations exist.
Implementation Scenario Example	• SAMBA allows Windows and UNIX machines to coexist and share files. • It can be implemented to replace Windows Domain Controllers for file and printer sharing, and authentication.

NFS	
Description	• NFS is a networked file system making the filesystem on a remote system accessible on the local system. From a user's perspective, an NFS-mounted filesystem is indistinguishable from a filesystem on a directly-attached disk drive. • Also considered to be the UNIX equivalent of the Server Message Block (SMB) protocol. • The original version was developed by Olaf Kirch and Alan Cox. The version 3 server code was solidified by Neil Brown, based on work from other developers. It is now in version 4.
Reference	• http://nfs.sourceforge.net/nfs-howto/
Rationale for selection	• Fast, seamless sharing of files over the network. Similar in features to SAMBA. • The advantage of NFS today is that it is mature, standard, well understood, and supported robustly across a variety of platforms.
Limitations	• No known limitations exist.
Implementation Scenario Example	• NFS allows different Operating Systems to share files. • It can be implemented together with SAMBA to provide filesystem access to users of the network.

CUPS	
Description	• The Common Unix Printing System (CUPS) is a modularized computer printing system for Unix-like operating systems that allows a computer to act as a powerful print server. • A computer running CUPS is a host which can accept print jobs from client computers, process them, and send them to the appropriate printer. • Originally created by Easy Software Products , it is available on all GNU/Linux and BSD type operating systems
Reference	• http://www.cups.org/articles.php
Rationale for selection	• CUPS is a widely used printing system • CUPS provides both the System V and Berkeley printing commands so the traditional methods of printing files can be used for CUPS. • The CUPS server itself runs a web server administration interface, so configuration is a relatively simple task. • The primary advantage of CUPS is that it is a standard and modularized printing system that can process numerous data formats on the print server.
Limitations	• No known limitations exist.
Implementation Scenario Example	• CUPS can be used in a heterogeneous computing environment.

2.5.9 Backup Server/Tool

Background description:

Any organisation requires a backup policy to ensure that critical data and applications are protected from system and infrastructure failure. The use of a backup server allows automated execution of this backup policy.

Amanda	
Description	• Amanda, the Advanced Maryland Automatic Network Disk Archiver, is a backup system that allows the system administrator to set up a single master backup server to back up multiple hosts to a single large capacity tape drive. • It was originally written by James da Silva while at the University of Maryland's Computer Science Department. • Today, AMANDA is completely maintained by a volunteer group, including a user community that provides most of the support.
Reference	• http://www.amanda.org/
Rationale for selection	• It is able to backup multiple servers on a network. • It can use SAMBA to backup windows servers. • It supports various backup devices, from hard disks to multi-tape libraries.
Limitations	• Does not have a native Windows agent to facilitate backup from Windows machines.
Implementation Scenario Example	• Amanda backup server can be used to backup files of data centers. • It was initially designed for UNIX and UNIX-like systems, with the ability to backup Windows machines reliant on the use SAMBA.

2.6 Network Security

2.6.1 Firewall

Background description:

A firewall is a program that resides at a network gateway server or external access point that protects the internal network from malicious users or packets from other networks.

iptables	
Description	• iptables is a firewall included in the Linux kernel version 2.4. and above. • iptables is maintained by netfilter.org.
Reference	• http://www.netfilter.org/
Rationale for selection	• iptables is able to do packet filtering, network address translation and other packet mangling. • It is able to add functionality by adding extensions. • It allows the creation of meta-rules to reduce the complexity of configuration. • It also has the ability to act as a bandwidth manager. • iptables is one of the most robust, and modern firewalls available today.
Limitations	• No known limitations exist.
Implementation Scenario Example	• In most cases it is implemented on network gateways. • It can also be implemented at various nodes in a network infrastructure to control flow of data across the network.

ipfilters	
Description	• ipfilter (commonly referred to as ipf) is a software package that can be used to provide network address translation (NAT) or firewall services. • IPFilter comes as a part of FreeBSD, NetBSD and Solaris 10. It can also run on GNU/Linux with kernel 2.4 and above. • Author and maintainer is Darren Reed. • ipfilter is able to explicitly deny/permit any packet from passing through.

ipfilters [cont'd]	
Reference	• http://coombs.anu.edu.au/~avalon/ or http://www.ipfilter.org
Rationale for selection	• Distinguish between various interfaces. • Provide packet header details to a user program for authentication. • In addition, supports temporary storage of pre-authenticated rules for passing packets through. • ipfilter is one of the most robust, and modern firewalls available today.
Limitations	• No known limitations exist.
Implementation Scenario Example	• In most cases it is implemented on the network perimeter. • It can also be implemented at various nodes in a network infrastructure to control flow of data across the network.

IPFW	
Description	• IPFW is a FreeBSD sponsored firewall software application. • It uses the legacy stateless rules and a legacy rule coding technique to achieve what is referred to as Simple Stateful logic. • IPFW is included in the basic FreeBSD install as a separate run time loadable module. • IPFW is authored and maintained by FreeBSD volunteer staff members.
Reference	• http://www.freebsd-howto.com/HOWTO/Ipfw-HOWTO
Rationale for selection	• IPFW is targeted at the professional user or the advanced technical computer hobbyist who have advanced packet selection requirements. • It is the user interface for ipfirewall and is also used as a tunnel shaper. • IPFW is similar to ipfilter but has advanced capabilities.
Limitations	• No known limitations exist.
Implementation Scenario Example	• It can also be implemented at various nodes in a network infrastructure to control flow of data across the network.

2.6.2 Network Intrusion Detection System (IDS)

Background description:

Network IDSes often act as a forensic tool for analysing network breaches. This is achieved by intercepting traffic entering and leaving the network, and analysing the packet headers and payloads. It requires active monitoring as well as constant updating of signatures to be effective.

Snort	
Description	• Snort is a open source network intrusion detection system, capable of performing real-time traffic analysis and packet logging on IP networks.
Reference	• http://www.snort.org/
Rationale for selection	• Snort has 3 distinct modes; sniffer, packet logger and intrusion detection. • It uses ACID (Analysis Console for Intrusion Databases) as a GUI reporting tool. • It has third-party GUI interfaces as well.
Limitations	• No known limitations exist.
Implementation Scenario Example	• The easiest way to implement Snort is by installing it at the gateways within a network. • It can also be installed behind the firewall as a second-level detector of network breaches.

2.6.3 Virtual Private Network (VPN)

Background description:

VPNs serve to reduce the cost of having secure internal network communications across geographically dispersed locations. Instead of having expensive dedicated leased-lines, an organisation can construct secure "tunnels" across the Internet, using strong encryption.

OpenVPN	
Description	• OpenVPN is a newer generation VPN in that it is based on SSL as the underlying security mechanism.
Reference	• http://openvpn.net/
Rationale for selection	• OpenVPN works on many different operating systems, including Linux, *BSD, Mac OS X and Windows. • It supports for dynamic IP addresses and NAT. • It fully supports OpenSSL PKI. • It uses an industrial-strength security model designed to protect against both passive and active attacks. • It has been rigorously designed and tested to operate robustly on unreliable networks.
Limitations	• No known limitations exist.
Implementation Scenario Example	• OpenVPN can be implemented in bridging or routing mode. Both modes have advantages and disadvantages. • A discussion on the two modes can be found at http://openvpn.net/faq.html#bridge1 • It is often installed between static remote machines. • It can also be installed to accommodate users who are often traveling, but yet need to be connected to their internal network.

2.6.4 Anti-virus

Background description:

Anti-virus software is designed to protect users from malicious programs such as viruses, trojans and worms.

CalmAV	
Description	• ClamAV is an anti-virus toolkit for UNIX, designed for e-mail scanning on mail gateways. • It provides a flexible and scalable multi-threaded daemon, a command line scanner, and an advanced tool for automatic database updating via the Internet and the virus signature database is kept up to date.
Reference	• http://clamav.net/
Rationale for selection	• ClamAV supports all popular operating systems such as Linux, BSD, MS Windows, AIX, Solaris & MacOS X. • It supports on-access scanning (Linux and FreeBSD only). • It detects over 30000 viruses, worms, and trojans, including Microsoft Office and MacOffice macro viruses. • It scans within archives and compressed files. • It has auto update on the virus database.
Limitations	• ClamAV does not disinfect infected files.
Implementation Scenario Example	• ClamAV is mostly used together with a mail transport agent (MTA) or any mail scanner to act as a mail gateway. • It can also be used as virus scanner on the desktop.

2.6.5 Anti-spam

Background description:

Anti-spam software is designed to detect and eliminate spam emails.

SpamAssassin	
Description	• SpamAssassin is a mail filter which attempts to identify spam using a variety of mechanisms including text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases. • These tests are applied to email headers and content to classify email using advanced statistical methods.
Reference	• http://spamassassin.apache.org/
Rationale for selection	• SpamAssassin is widely used in all aspects of email management. • It has a modular architecture that allows other technologies to be quickly wielded against spam and is designed for easy integration into virtually any email system. • Its practical multi-technique approach, modularity, and extensibility continue to give it an advantage over other Anti-spam systems. • It is ready in use in both email clients and servers, on many different operating systems, filtering incoming as well as outgoing email. • It has been shown to produce around 0.9% false negatives (spam that was missed) and around 0.1% false positives (incorrectly marked as spam).
Limitations	• On accuracy, SpamAssassin typically differentiates successfully between spam and non-spam in between 95% and 100%.
Implementation Scenario Example	• SpamAssassin is mostly used together with mail transport agents (MTA) or any mail scanner to act as a mail gateway.

3 RECOMMENDATIONS FOR INFORMATION ACCESS

3.1 Overview

Information Access within MyGIFOSS covers components and technical specifications which are not specified in, or have not been considered under, MyGIF v1.0. MyGIFOSS takes precedence over MyGIF for OSS implementations.

Information Access covers components and technical specifications required to enable users to access Public Sector information and services electronically via a range of delivery channels (e.g. World Wide Web) and devices (e.g. personal computers, mobile phones, PDAs). Interoperability components covered in the Information Access area include:

- Hypertext Web Content;
- Document;
- Spreadsheet;
- Presentation;
- Graphical Image;
- Moving Image and Audio / Visual Content;
- Audio / Video Streaming;
- Animation;
- Mobile Devices Content;
- Character Sets and Encoding;
- Compression; and
- Client-Side Scripting.

It should be noted that standards can, and do, evolve. In addition, standards are also created and deprecated continuously. Therefore, the standards defined here are non-exhaustive, and will be constantly updated.

3.2 Recommended Standards / Specifications

3.2.1 Hypertext Web Content

Hypertext Web Content standards are required to specify the development and formatting of hypertext documents for presentation on browsers via a range of delivery channels including Internet and Intranet.

Recommended standards and specifications are as defined in MyGIF.

3.2.2 Document

Standards on Documents are required to define the format and file types of documents for interchange between agencies and departments as well as third parties.

Recommended standards/specifications:

- OASIS Open Document Format for Office Applications v1.0

Other recommended standards and specifications for documents are as specified in MyGIF, with the exception of the Microsoft Office Word Document (.doc) format. This is because the format is not fully portable to other platforms and word processing software.

OpenDocument Text Format (.odt)	
Description	• The OpenDocument Text Format is part of the OpenDocument for Office Applications standard, specifically aimed at word processing.
Reference	• http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
Rationale for selection	• The standard defines an XML schema which is well documented and published. • It is friendly to transformations using XSLT or similar XML-based tools. • OASIS has submitted the OpenDocument Format OASIS Standard to the ISO/IEC JTC1 (International Organization for Standardization International Electrotechnical Commission's Joint Technical Committee) for further approval as a de jure standard. • It has the ability to open and write to Microsoft Word .doc formats 97/00, 95 and 6.0.
Limitations	• Incompatibilities with Microsoft Word may exist (due to newly introduced proprietary features by Microsoft).

3.2.3 Spreadsheet

Standards on Spreadsheet are required to define the format and file types of spreadsheets for inter change between agencies and departments as well as third parties.

Recommended standards / specifications:

- OASIS Open Document Format for Office Applications v1.0

Other standards and specifications for spreadsheets are as specified in MyGIF, with the exception of the Microsoft Office Excel Spreadsheet (.xls) format. This is because the format is not fully portable to other platforms and spreadsheet software.

OpenDocument Spreadsheet Format (.ods)	
Description	• The OpenDocument Spreadsheet Format is part of the OpenDocument for Office Applications standard, specifically aimed at word processing.
Reference	• http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
Rationale for selection	• The standard defines an XML schema which is well documented and published. • It is friendly to transformations using XSLT or similar XML-based tools. • OASIS has submitted the OpenDocument Format OASIS Standard to the ISO/IEC JTC1 (International Organization for Standardization International Electrotechnical Commission's Joint Technical Committee) for further approval as a de jure standard. • It has the capability to read and write other file formats including Microsoft Office Excel .xls (97/00, 95, 5.0).

OpenDocument Spreadsheet Format (.ods) [cont'd]	
Limitations	• Features unique to Excel 97/2000/XP may not be displayed the same way in OpenOffice. For instance, a workbook that contains preformatted spaces in Microsoft Excel may show up but the spaces might be different. • Interpretation of formula can be different between .odc and .xls files, though this is increasingly rare. • Visual Basic macros that use commands new to Excel 2000 and XP may result in execution errors when run in OpenOffice.

3.2.4 Presentation

Standards on presentation are required to define the format and file types of presentation for interchange between agencies and departments as well as third parties.

Recommended standard / specifications:

- OASIS Open Document Format for Office Applications v1.0

Other standards and specifications for presentations are as specified in MyGIF, with the exception of the Microsoft Office Powerpoint (.ppt) format. This is because the format is not fully portable to other platforms and presentation software.

OpenDocument Presentation format (.odp)	
Description	• The OpenDocument Presentation Format is part of the OpenDocument for Office Applications standard, specifically aimed at word processing.
Reference	• http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office
Rationale for selection	• The standard defines an XML schema which is well documented and published. • It is friendly to transformations using XSLT or similar XML-based tools. • OASIS has submitted the OpenDocument Format OASIS Standard to the ISO/IEC JTC1 (International Organization for Standardization International Electrotechnical Commission's Joint Technical Committee) for further approval as a de jure standard. • It has the capability to read and write other file formats including Microsoft Office Excel .xls (97/00, 95, 5.0).
Limitations	• Formatting data may be lost when converting to/from Microsoft Powerpoint.

3.2.5 Graphical Image

Standards on graphical images are required to define the format and files types of graphics and still images for interchange between agencies and departments as well as third parties.

Recommended standards and specifications are as contained within MyGIF with the following additional information and format additions:

- Joint Photographic Experts Group (.jpg)
- Graphic Interchange Format (.gif)
- Tag Image File Format (.tif)
- Portable Network Graphics (.png)
- XpixMap(XPM) (.xpm)

Joint Photographic Experts Group (.jpg)	
Description	• Joint Photographic Experts Group (JPEG) is an ISO graphic image file format standard (ISO918).
Reference	• http://www.jpeg.org/
Rationale for selection	• JPEG standard is widely supported by browsers and the majority of image processing, graphics designing, photo processing and scanner accessory software. • It is a mature standard – originally ratified in 1994 and is natively supported by Mozilla, Firefox and Konqueror. • It is capable to save images with millions of colors.
Limitations	• The JPEG format is under patent litigation threat by a company called Forgent Networks, for its compression algorithm. However, many commercial providers of applications and devices which use JPEG have licensed the patent.

Graphic Interchange Format (.gif)	
Description	• Graphic Interchange Format (GIF) is one of the most common formats for graphics images on the Web.
Reference	• GIF v89a is a standard defined by CompuServe Incorporated and available at: http://www.w3.org/Graphics/GIF/spec-gif89a.txt
Rationale for selection	• Graphic Interchange Format is a de-facto standard widely supported by browsers and the majority of image processing, graphics design, photo processing and scanner accessory software. • It is natively supported by Mozilla, Firefox and Konqueror. • It can be animated. • Formerly had a patent on its compression algorithm, but has since expired (June 2003).
Limitations	• GIF only provide 256 color patterns.

Tag Image File Format (.tif)	
Description	• Tag Image File Format (TIFF) was developed by Aldus and Microsoft Corp, and the specification was owned by Aldus, which in turn merged with Adobe Systems, Incorporated. Consequently, Adobe Systems now holds the Copyright for the TIFF specification. • TIFF is a common format for exchanging raster graphics (bitmap) images between application programs. It is a de-facto standard of particular benefit for images that will not tolerate information loss.
Reference	• TIFF version 6 specification is available at http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf
Rationale for selection	• TIFF is a de-facto standard of particular benefit for images that will not tolerate information loss. • Version 6 is the current version and a matured standard. It was published in June 1992. • It is widely supported by browsers through freely-available plug-ins and the majority of image processing, graphics design, photo processing and scanner accessory software. • Formerly had a patent on its compression algorithm, but has since expired (June 2003).
Limitations	• No known limitations.

Portable Network Graphics (.png)	
Description	• Portable Network Graphics (PNG) is an extensible file format for the lossless, portable, well-compressed storage of raster images.
Reference	• http://www.libpng.org/pub/png/pngintro.html/
Rationale for selection	• For the Web, PNG has three main advantages: alpha channels (variable transparency), gamma correction (cross-platform control of image brightness), and two-dimensional interlacing (a method of sieve display). • For image editing, either professional or otherwise, PNG provides a useful format for the storage of intermediate stages of editing. • It is supported by most browsers, with the most notable exception being the proprietary web browser, Internet Explorer – it does not support PNG without external plugins.
Limitations	• No known limitations.

XpixMap (.xpm)	
Description	• XPixMap (XPM) consists of an ASCII image format and a C library. • The format defines how to store color images (X Pixmap) in a portable and powerful way.
Reference	• http://koala.ilog.fr/lehors/xpm-README.html
Rationale for selection	• While XPM is not an X Consortium standard, it is already a de-facto standard. This has been used widely by both commercial and non-commercial developers. • This is the format of choice when simplicity is required.
Limitations	• No known limitations.

3.2.6 Moving Image And Audio / Visual Content

Standards on Moving Image and Audio/Visual contents are required to define the compressed format and file types of audio/visual content such as movies, for interchange between agencies and departments as well as third parties.

Recommended standards and specifications are as defined in MyGIF.

3.2.7 Audio / Video Streaming

Audio/Video Streaming Standards on Audio/Video Streaming are required to define the formats and file types streaming audio/video content such as web casts and web seminars, for interchange between agencies and departments as well as third parties.

Recommended standards / specifications:

- Helix DNA
- VideoLAN

Helix DNA	
Description	• Helix DNA is an open multi-format digital media platform, intended to accelerate the development and deployment of digital media applications of any media, on any device, over any network, on any operating system and created in any development environment. • Helix originated from Real Networks' Real Audio/Video streaming server and player. It was made open source by Real Networks in 2003.
Reference	• http://helixcommunity.org/
Rationale for selection	• The Helix DNA system supports multiple media formats, including Real Media, Windows Media and Quicktime. • The Helix Server and Player are cross-platform with clients across all major operating systems.
Limitations	• No known limitations.

VideoLAN	
Description	• VideoLan is a cross-platform media server and player. • It started as a student project at the French École Centrale Paris, then released under the GPL. • The VideoLan server can stream MPEG-1, MPEG-2 and MPEG-4 files, DVDs, digital satellite channels, digital terrestrial television channels and live videos on the network in unicast or multicast.
Reference	• http://www.videolan.org/
Rationale for selection	• It supports wide range of audio/video formats and codecs. • Available on multiple operating systems.
Limitations	• No known limitations.

3.2.8 Animation

Animation standards are required to define the applications and formats to be used for the inter change of animated content between agencies and departments as well as third parties.

Due to the lack of mature OSS animation formats, standards and specifications as defined within MyGIF can be used. Users are however advised to use publishing software which provide players that are available on OSS platforms, for e.g. Macromedia's Flash.

3.2.9 Mobile Devices Content

Mobile Devices Content Standard on Mobile Device Content is required to define the format of content for presentation on mobile devices such as mobile phone and PDAs.

Recommended standards and specifications are as defined in MyGIF.

3.2.10 Character Sets And Encoding

Character Sets and Encoding standards define the character sets to be used for content to be interchanged in English or Malay, as well as how those characters are to be encoded.

Recommended standards and specifications are as defined in MyGIF.

3.2.11 Compression

Compression is required to define the applications and format to be used for compressing files for interchange in between related parties.

Recommended standards / specifications:

- Zip
- GNU Zip
- Bzip

Zip (.zip)

Description	• Files in a zip file are compressed so that they take up less space in storage or take less time to send to someone.
Reference	• Zip specification is available at: http://www.pkware.com/products/enterprise/white_papers/appnote.htm
Rationale for selection	• Zip is the de-facto standard for file compression. • It is a global, matured and widely adopted standard. It was introduced in 1989. • It is supported on a range of operating systems including Linux, BSD, UNIX and Windows. Extractors are freely available.
Limitations	• No known limitations.

GNU Zip (.gz) v4.3

Description	• GZIP (GNU zip) is a compression utility. • It has been adopted by the GNU project and is popular on the Internet.
Reference	• GZIP is an IETF standard defined by RFC 1952 “GZIP file format specification version 4.3”: http://www.gnu.org/
Rationale for selection	• GZIP is a commonly utilized file compression format. • It is supported on a range of operating systems including DOS, UNIX and MacOS. It can be extracted on Windows operating systems using the freely available WinZip utility.
Limitations	• No known limitations.

bzip2 (.bz2)

Description	• bzip2 is another file compression utility. • The bzip2 utility are newer than gzip and is not as common yet, but it is rapidly gaining popularity.
Reference	• Information on the use of this type of compression can be obtained at: http://www.linuxheadquarters.com/howto/basic/bz2.shtml
Rationale for selection	• The bzip2 utility is capable of greater compression ratios than gzip. • Therefore, a bzip2 file can be 10-20% smaller than a gzip version of the same file. Usually, files that have been compressed by bzip2 will have a .bz2 extension. • The use of this format of compression is compatible with a wide range of Operating System such as Linux, Windows and UNIX.
Limitations	• No known limitations.

3.2.12 Client-Side Scripting

Client-side scripts are programs written and attached or embedded to HTML documents in a manner independent of the scripting language. The scripts add interactivity and program logic to browser-based content, for instance, provide runtime validation of form field contents by responding to a user's mouse action with the execution of program to validate user input. It reduces server load by transferring some of the processing of the program to be handled locally at client.

Standard on client-side scripting is required to ensure consistency on the scripts implementation at different browsers, in particular, the dominant browsers such as Firefox, Mozilla, Konqueror, Microsoft IE and Netscape Navigator.

Recommended standards and specifications are as defined in MyGIF.

PART B

4 IMPLEMENTATION GUIDELINES: MANAGEMENT PERSPECTIVE

Using OSS creates new challenges to be addressed. On the other hand it also gives opportunities to re-engineer your IT System. OSS allows a paradigm shift from the conventional methods of IT services. It is a move away from a product oriented industry to a service oriented Industry. What managers should be asking themselves are: how to ensure interoperability of their system; how to support mobile users; how to securely identify remote users and how to build a manageable IT system. OSS installation cost is minimal or almost none. However, the main issue is support, and that is why IT managers should understand the dynamics of Open Source and realize the importance of community involvement into getting support from vendors and open source communities. This chapter assists IT managers who are planning for an implementation or a migration project in OSS.

4.1 Key Success Factors

The following are critical for the successful implementation of an OSS project:

I. **Understanding the reasons to migrate before starting the Project**

The need for open standards in electronic government, standardization for data exchange, the higher level of security in OSS, the elimination of vendor lock-in and forced upgrades by vendors and lower cost are some of the reasons to change to OSS.

IT managers should therefore internalise the reasons for their migration and analyze whether it is aligned with their goals and objectives.

II. **Acquiring positive support from IT staff and users throughout the project**

In any move from the norm, support is important. Hence, IT managers should gather all the support for an OSS implementation to be successful. This primarily concerns users and IT staff, since they are the ones who will be evaluating, applying and using OSS.

III. **Obtain top management commitment as owners of the project and championing the change**

Top management plays the biggest role in change. Acquiring top management support and guidance will assure continuous commitment to the project. This will also assure sufficient human resources, funding and training.

IV. **Building relationship and expertise with other OSS movements**

OSS development are largely community driven, though increasingly with vendor involvement and support. Therefore, it is beneficial to be involved in the community, since this will give an insight into the software development cycle, as well as allow close communications. This will result in greater internal expertise, and better support from the community and vendors.

V. **Develop an initial proof of concept**

Depending on the reasons to migrate to OSS, it may be important to first aim small by targeting achievable goals on pilot projects. As a proof of concept, this will reinforce your business case in your overall OSS planning. This will also help in providing assurance to sceptics when moving to the actual implementation later.

VI. **Manage each step of implementation to assure viability**

Planning must cover areas such as cost analysis, change management, human resource management, training, and benefit validation and realization.

Based on the above areas, manage the implementation with proactive and reactive measures so that the viability of the implementation can be assured.

4.2 Challenges

While implementing an OSS project, an IT manager may face multiple challenges that might jeopardise the project. These challenges can possibly be less potent over time as they are mitigated by positive actions from the OSS communities, vendors and other involved parties.

However, the following are items that a manager should be aware of as potential challenges:

No	Area	Challenges Faced
1	Technology	• The perception of OSS security & authentication needs to be addressed. • Requirement for upfront investment to be provided for initial OSS implementation. • Dependency on applications from proprietary IT companies makes it difficult to implement alternatives, resulting in vendor lock-in. • Limited or non-existence of readily available OSS equivalent solutions to meet some business needs. • Non-OSS compliant IT peripherals (printer, scanner). • Lack of specific off-the-shelf OSS-based applications especially for education. • Interoperability and incompatibility of current data and file formats.
2	Human	• Resistance to change among users. • Lack of management buy-in. • Insufficient number of in-house skilled personnel in OSS for technical support. • Limited external technical support (both for developers and users) available for the development and deployment of OSS solutions. • Lack of an active OSS community within the organisation.
3	Policies and procedures	• Lack of OSS proponents in the tender evaluation process at the agency level. • Lack of understanding of the concepts of Intellectual Property, specifically Copyright, Patents and Trademarks give rise to confusion with regards to the legal standing of OSS.

4.2 Challenges [cont'd]

No	Area	Challenges Faced
4	Organisations	• Lack of awareness, understanding and confidence in Open Source initiatives at all levels in the organization. • Mindset and cultural acceptance of OSS needs to be addressed at all levels in the organization. • Lack of R&D initiatives, incentives and recognition programmes for the internal IT personnel and OSS community. • Collaboration efforts among organisations to spur knowledge sharing are not fully optimised.
5	Other Factors	• Inadequate allocation of budget and funds. • Lack of studies on benefits and new products available in OSS. • Ready availability of pirated software and lack of enforcement allows the use of illegally obtained copies within the organisation.

Table 4.1: Potential Challenges in Implementing OSS

4.3 Project Management Of An OSS Implementation

IT Managers should consider the following factors:

No	Phase	Description	Considerations
1	Data gathering and project definition	Sets of relevant initial conditions (Current environment)	System Architectures
			Application and data associated with it
			Protocol and standard used
			Hardware Used
			Physical Infrastructure like location, networks, bandwidth
		Set of target Condition	System Architectures
			Applications and data associated with it
			Protocols and standards used
			Hardware Used
			Physical Infrastructure like location, networks, bandwidth
		Implementation Method	Project methodology
			Project life cycle
2	Justification of OSS Project	Tangible benefit for implementation	Cost associated to project
			Ease of Operations
			Free from Vendor lock-in
		Intangible benefit for implementation	Cost associated to project
			Ease of Operations
			Free from Vendor lock-in
3	Pilot Testing	One or more pilots of the project can be implemented	Consider the cost in the model and apply it in the whole project plan
			Consider the time line and resources used in the pilot projects and feed the variables into the overall project implementation
4	Project Plan Roll Out	Follow on with the project implementation after modifications from the Pilot Testing Phase	Consider whether the scaling of the project from pilot to roll out is realistic and make adjustments and accommodations, if necessary

4.3 Project Management Of An Oss Implimentation [cont'd]

No	Phase	Description	Considerations
5	Monitor	Keep monitoring throughout the project changes and discrepancies with respect to the original plan	Consider a plan that are more flexible and capable of changing to unforeseen circumstances

Table 4.2: Considerations for OSS Project Management

4.4 Notes For Migration

For successful migration to OSS, it is advised that the following guidelines are followed:

I. Steering committee

A steering committee comprising relevant senior management personnel from all relevant departments should be formed. The committee will provide guidance and direction for the implementation.

II. Understanding the target environment

OSS software and the base architectures should be fully understood, and options of variations and choices available should be analyzed. Usually this requires the training of the existing work force, acquisition of new staff with new skill sets or using outside expertise. Here is where management support is important since a considerable amount of investment must be done.

III. Opportunity to re-engineer the current system

Multiple approaches in implementing a system creates opportunities to re-engineer the current base architecture and its software applications. Consider an appropriate approach to centralized or decentralized management control. However, careful considerations should reflect cost related to such changes.

IV. Understand OSS

There are a few factors that need to be understood before OSS migration. After sufficient knowledge about OSS is obtained and understood, only then considerations about OSS can be made properly. These factors are, for example:

a. The implications of OSS licenses, should the project owner make changes to the original source code and wishes to distribute it.

b. The advantages and disadvantages of various flavours of an application to be implemented.

c. The various distinct characteristics of Open Source distributions and the vendor support that it provides.

d. The level of support that exist for a particular OSS application. For example, commercial, community or third-party vendor support.

V. IT system environment scanning

Audit the current system to formalize a precise cost analysis for a business case. Data gathered should be for the following:

A. Application requirements

i. Application name, Vendor Name, Version Number and Contact Person.

ii. The number of users using the application.

iii. The operating system used and alternative to using other OS or possibility of work-around with OSS.

iv. Application prerequisites on the client-side, server side as well as middle-ware to support the application.

v. Hardware requirements of the application.

vi. Communication protocol and data access methods for the applications to work with other applications.

vii. File format used by an application.

viii. Localization and internationalization requirements such as currency, language and time.

B. Data requirements

This should include all data in the system such as file formats, information exchange format, application format and protocols, databases and any other application that is interacting with the computer system.

The analysis of the requirements should include considerations such as:

i. The need for interacting with the system by a third party organization or software application.

ii. The requirements to persist on legacy applications due to critical data or historical data.

One of the way is to divide the data into the following categories:

i. Data of minimal importance and impact such that it can be thrown away.

ii. Data of critical importance and must be preserved either in its original form or translated to an open standard form.

iii. Data of critical importance and must be preserved and maybe be only viewed by a proprietary system.

The criticality of the data, and the need for external interaction would have an impact in the cost for translating them, or in maintaining a proprietary application.

C. Security requirements

i. The current system methods to handle authentication with users and passwords.

ii. The authentication structure as of name formats, conventions and others.

iii. Policy for user management, updates, adding and deletes.

iv. Requirement of other authentication systems that are more complex for example, thumb printing, retina scan, signature identification and smart card requirements.

v. The IT policy regarding the use of Internet access like email, messenger, browser, peer-2-peer file sharing and etc.

vi. Security requirements that are dependent on specific hardware or software vendor.

VI. Build a detailed business case

Based on the data from the system environment scanning, a detailed business case for migration can be formulated and should at least consist of the following details:

a. PESTLE or SWOT analysis of the new environment compared to the existing and alternative environments.

b. Total Cost of Ownership (TCO) projection over a reasonable period of time. This should include the costs of data migration from existing applications, software acquisition, training, support and maintenance. A 5 year plan is recommended but it is primarily determined by the agency and their IT project life cycle.

c. A TCO projection if the current system is maintained, or migrated/upgraded to another proprietary system. The factors used in the TCO projection should be the same as above, within the same period of time.

d. Cost comparisons between the different TCO projections.

VII. Use a consultative process

Include all stakeholders in the migration, which means the management, IT implementation personnel, end users and external vendors in the planning. This includes making them understand the migration purpose and implications to the user. Furthermore, users' concerns should be taken seriously. This is important to assure smooth transition and early user involvement in the project. Throughout the project, provide a help desk to accept concerns from users as well as provide how-to's to working solutions.

VIII. Start small

Implement a pilot project based on the business case for the migration. Consider achievable targets with small number of users and small isolated cases. Apply any changes to the business case where required. The pilot will provide:

a. Data to modify the TCO model.

b. Data to modify change management methods by analyzing user responses to the pilot.

c. Proof of concept for the business case.

IX. Determine the method and speed of migration

Usual options are:

a. Big bang

A big bang approach consists of a mass migration of all the users onto the new system with little or no transition period. In certain cases, this can be the most cost efficient method, since it would not require maintenance of the legacy system. However, this approach can involve a lot of preparatory ground work and would require all data to be fully migrated and online first.

b. Phased transitions in groups

A more common approach is to partition the organisation into groups, for e.g. by divisions, departments or work area. Usually, the IT department would be the first to migrate, since the personnel would be the most comfortable working with computers. This would then be followed by other departments or groups, for example Administrative, Accounting, Secretarial, Policy & Planning, and so on.

This approach has the advantage of allowing for a more staged change management program. It also allows early issues in the system to be ironed out.

c. User by user transition

This is the slowest and most resource intensive approach, since it entails implementing the migration on a per user basis. This approach can prove to be unimplementable in large organisations.

X. Promotion and communication

It is important that the users are aware of the roll-out throughout the organisation. Therefore, sufficient promotion of the migration process needs to be broadcasted. Further, the affected users must be constantly informed to the changes being undertaken within their organisation. By utilising effective communication, users will be well informed and receptive of the changes.

XI. Monitor and obtain feedback

Any implementation may have oversights. It is then imperative that these are identified, and rectified quickly. Further, users involved in the implementation would often have valuable suggestions and ideas with regards to improving the implementation. Obtaining and acting on these feedback ensures better results and more commitment from the users.

4.5 Common Change Management Issues

A migration process involves change on several levels, not just at the computer systems and infrastructure. Change management involves handling the challenges at all levels, especially people. The following discusses some of these change management issues.

I. Fear of the unknown

Naturally, human beings dislike change or anything that is different from the norm. Trying out something new like OSS will definitely cause some resistance from people who are adverse to change.

This is where people management and communication skills are important, for a smooth transition to occur. This can be done by listening carefully to people's fears, and acting swiftly to calm those fears.

II. Inertia

In some cases, there is no real fear of the unknown, but just a general unwillingness to accept something new. This is mainly due to being comfortable with what is already available.

III. Opposition to change

Fear and inertia may cause users to resist change. It may also induce them to actively oppose the change. This is especially difficult when those opposed to change are in a position to affect the migration process. Therefore it is important to communicate to them the reasons behind the migration and to convince them to actively support it instead.

IV. Less demand in job skill set

Almost all users are afraid of change, especially if they are required to be reskilled. Reskilling of users would not be so difficult if the users believe that their newly acquired skill set would be valuable in the future. However, OSS yet is not as popular as other proprietary products which are already well established. This gives the perception to users that learning OSS can make them less valuable. Management must handle this issue carefully and understand the pressure and the concerns of the users. With the gaining popularity of OSS, this issue would gradually lessen.

4.6 Making The OSS Transition Easier

OSS introduction can be difficult both on technical and management point of view. Below are points that could possibly make it easier for introducing OSS:

I. Take the least disruptive route first

Regardless of whether the implementation approach is a big bang, phased or on a per user basis, the introduction of OSS into the organisation should be taken step-by-step. Hence, the administration should start introducing OSS in areas which users are fairly familiar. Applications like OpenOffice, Apache, Gimp and Mozilla Firefox are good examples of applications that can be used in the early stages of introducing OSS since it has functionalities similar to their proprietary counterparts and are mostly capable of running on various operating systems.

The next step is to introduce least disruptive applications and servers. These are applications like DNS servers, DHCP servers, FTP servers, LDAP servers, mail servers and other applications that are almost transparent to the users in the system.

II. Technical consideration

Looking ahead is essential to avoid making decisions that would be difficult to undo later. Therefore, there are several factors which the implementation team should consider before rolling out to avoid major stumbling blocks in the future.

Recommended	Discouraged
Web development done today are compliant with most browsers. Essentially, web development should follow W3C standards. Tools like weblint can help test web pages for compatibility.	The use of Vbscripts, active-X and scripts that are platform specific to be used in Web development. This is because those languages and tools create applications that are not compatible with all browsers.
Different ways to do specific functions on applications that require the use of proprietary languages.	The use of Macros and VbScripts in documents like Microsoft Excel and Microsoft Access. This is because those macros are not fully portable and also can act as carriers for viruses.
Open standards for file formats, for example, OpenDocument.	The use of proprietary file formats like Excel, Powerpoint, Word and others that require a vendor specific application to be used. This is because those formats are not fully portable.
Open standard protocols. Refer to ***Chapter 2*** and ***Chapter 3*** of MyGIFOSS to look at open standard protocols and applications.	The use of proprietary protocols like proprietary encryption protocols, communication protocols and messaging protocols. This is because proprietary protocols tend to be platform specific and non-portable
All new applications are written to be portable by using cross platform languages like JAVA, Python, Perl and ANSI C.	Applications that can run on proprietary platforms only.

Table 4.3: Technical Consideration

III. OSS Implementation Checklist

A checklist on OSS implementation is provided below and should be filled up when constructing your implementation plan.

No	Item	Yes/ No
1	Registration at the OSS Knowledge Bank	
2	Attendance to the OSS Awareness program	
3	Completed OSS Questionnaire	
4	Availability of current environment: • ICT environment • ICT Governance • ICT resources skills and expertise • ICT Policies • ICT Funding availability • Implementation Constraints	
5	Identified OSS Opportunities	
6	Validated OSS Opportunities	
7	Accurate assessment of the existing environment & OSS Opportunities	
8	Technical Feasibility Check	
9	Budgetary estimations and favourable Return on Investment (ROI) projections	
10	ICT Governance Approvals, Selection and Award to Vendor / Internal Resources for Implementation	
11	Detailed Implementation Plan	
12	Availability of Training Plan	
13	Availability of Change Management Plan	
14	Availability of Risk Management Plan	
15	Availability of separate development and testing environments	
16	Availability of Data Centre planning	
17	Availability of clear business and technical user requirements	
18	Definition, selection and assessment of OSS technical environment related to: • hardware • software (OS and applications) • tools inclusive of sources for signature updates for security • transition strategy • policies and procedures	
19	Availability of skilled OSS resources	
20	Availability of User Awareness	
21	Availability of Technical and User training	
22	Availability of Maintenance and Support processes and procedures: • user support • bug fixes and patches • updates / upgrades	
23	Realisation of Benefits	
24	Show Case documentation deposited in Knowledge Bank	

Table 4.4: OSS Consultation Implementation Checklist

5 IMPLEMENTATION GUIDELINES: LEGAL AND TECHNICAL PERSPECTIVE

5.1 Overview

The implementation guidelines contained herein addresses the issues of using OSS and proprietary software within a heterogeneous environment, and technical guidelines for OSS implementation and migration.

5.2 Use Of OSS And Proprietary Software In A Heterogeneous Environment

Usage of mixed software within an operating environment can be looked at from two aspects:

I. Legal

From a legal perspective, usage of software are under the licensing terms and conditions of the software – which governs the rights of the end user on the usage of the software. A full treatment of the legal aspects with regards to software usage is beyond the scope of this document. However, one specific legal scenario relating to the GNU General Public License is given here as a brief example. This is discussed in ***Section 5.3***.

The scenario discussed in the section is designed to provide IT managers with a general idea on how licensing affects software implementation. The full implication of OSS licenses and implementation scenarios are contained in the ***OSS Policy Handbook***.

II. Technical

From a technical perspective, as long as the software is in a format which is understood by the operating system and underlying hardware architecture, then there would be no difficulties in running the software. The technical options for implementation, regarding the use of OSS and proprietary software are discussed in ***Section 5.4.***

5.3 Licensing

The following two subsections discuss the use of software licensed under the GNU General Public License with software under other licenses. This particular license is chosen as an example since it is the one that generates the most questions with regards to usage, and is also the license for the Linux kernel.

5.3.1 Differences Between Modification And Usage Of Software

OSS licenses generally grant the user certain rights to using the software, which include the right to use, modify and distribute the software. Certain licenses, for example the GNU General Public License, impose certain obligations on the modification of the software. There are many interpretations on these obligations, but the clearest one can be described as follows:

- The software can be modified and redistributed, provided the same rights granted to the modifier is granted to the recipients. In other words, any code that modifies GPL code or includes GPL code, or any portion of a GPL'd software, cannot be distributed *outside your organisation* under any license other than the GPL.
- This particular obligation is restricted to *modification* only.
- Rights governing the *usage* of the software, i.e. running it, do not impose the same obligation as to the distribution of the modified versions. This is an important distinction as will be made clear in the discussion to follow.

5.3.2 Example Scenario Of A Heterogeneous Software Environment

Consider the software stack example shown in ***Figure 5.1***. It shows a typical example of an application system, consisting of several subsystems. The lowest layer of the stack is the hardware, followed by the operating system. It then increases in the levels of abstraction from the hardware layer to the highest stack, which is the top application layer. This layer is the application that interacts with the user.

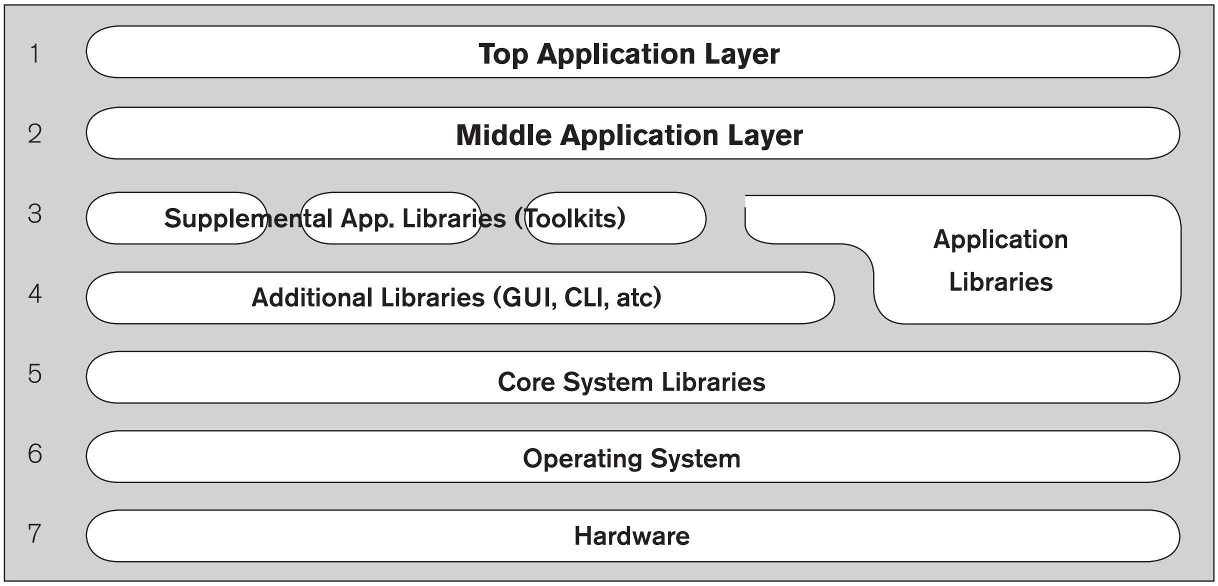

Figure 5.1 Example of a Software Stack

On any layer of the stack (excluding the hardware layer), the software can be proprietary or OSS. There are many scenarios to describe the interaction between the layers of software, but for illustration, a single one is described here.

Consider a Linux-based operating system, as provided by a distribution vendor (erg. Red Hat, SuSE, Debian, Slackware, etc.). Here, the operating system kernel (layer 6 above) is Linux, licensed under the GPL. The main core library (layer 5) is the GNU C library, licensed under LGPL . The upper layers can contain any number software, licensed under different licenses.

Next, consider the case of a proprietary database, which would slot in the middle application layer (layer 2). If the database is not linked (dynamically or statically) to any GPL'd (or similarly licensed) libraries, and do not depend on any other GPL'd (or similarly licensed) software to run, then the database can remain proprietary and closed source. Examples of databases that run on Linux which are proprietary are:

i. Oracle Database
ii. IBM DB2
iii. SAP DB

Other scenarios can include the use of OSS within a proprietary operating system environment. For example, the list below give a representation of OSS that run on the proprietary Microsoft Windows operating system:

i. Apache Web Server
ii. Mozilla Web Browser
iii. OpenOffice Productivity Suite

The following subsections in this section go into further detail on the usage of OSS and proprietary software.

In summary, the following points hold true:

i. Subject to the software's licensing terms, running unmodified software in an heterogeneous environment do not represent any obligations on the users to release the source code.

ii. Use of *modified* OSS within an organisation (in this case the Government of Malaysia) do not represent any obligations on the users to release the source code. The obligations come into effect should the software be distributed *outside the organisation*.

iii. The Government of Malaysia constitutes several different organisations which are legal entities in their own right. For example, State governments are legally separate organisations to the Federal government, and to each other. Therefore, the distribution of software from one state to another, or to the federal government (and all other exchange permutations) will be considered as distribution of the software. As such the obligations would apply.

5.4 Technical Guidelines For Usage Of OSS

The following subsections discuss technical options for:

i. Proprietary applications which have an OSS equivalent,
ii. Proprietary applications which run in an OSS environment,
iii. Software which may be accessed by remote display,
iv. Software which will run under an emulator, and
v. Software which can be recompiled under OSS.

A large proportion of this discussion is based on the European Commission IDA Migration Guidelines for OSS, which can be found at:

http://www.netproject.com/docs/migoss/v1.0/

The IDA Migration Guidelines also contains several migration scenarios which are very detailed, in respect to the steps needed. The reader is encouraged to refer to the link above as an additional supplement to this document.

5.4.1 Proprietary Applications Which Have An OSS Equivalent

Some applications, for instance Microsoft Office, Lotus SmartSuite, Corel WordPerfect, Adobe Framemaker, Quark Express and Adobe Photoshop, have equivalents which run natively under OSS, including OpenOffice.org, Gnumeric, Evolution and The GIMP. In this case the OSS product needs to be tested to ensure it provides the necessary functionality. Refer to ***Table 2.1*** for examples of OSS software within the categorised implementation and solution areas.

5.4.2 Proprietary Applications Which Run In An OSS Environment

Some applications, such as Adobe's Acrobat Reader, have a version which runs natively under OSS. If there is no OSS alternative to the application, then all that is required is to ensure that all the required features are implemented in the proprietary version. If there is an OSS alternative and a partial migration is acceptable, then a choice has to be made based on the features offered by the proprietary and OSS applications.

5.4.3 Software Which May Be Accessed By Remote Display

Another approach is to run the applications on a server and transport the display to the desktop; this is the thin client approach. Products like Windows Terminal Server, Citrix and Graphon allow applications to run on a server running Windows in a multi-user way. This means that an application written to run on the desktop in a single user mode may have to be altered to run under these products. This will not be possible without source code, and third party vendors may not be willing to help.

A comparison of various terminal server products can be found at:

http://www.networkcomputing.com/technologies/infrastructure/showitem.jhtml?articleID=57703288

The most sophisticated of these products, Citrix, has its own line protocol, "ICA", which is extremely good, particularly with low bandwidth connections. It can run a server farm with load balancing and has other useful facilities. There are free ICA clients which run under GNU/Linux.

All these products rely on proprietary closed source software and Citrix in particular is expensive. It requires a Windows server license, a Citrix license and a Windows Terminal Server license if a non-Windows client is used. In addition, Client Access Licenses will be needed for each desktop using the software. The Citrix license is based on concurrent users, so this approach can be cost effective if there are many users who need access to an application but where concurrent access is low. There are documented case studies at http://www.citrix.com/press/news/profiles/ which show that the savings of having thin client "disposable" desktops are sufficient to justify moving the applications to a server. Citrix also has products to allow UNIX applications to be transported in the same way over ICA and displayed on a thin client desktop.

Windows Terminal Server provides similar functionality to Citrix except that it uses its own protocol, RDP. The GNU/Linux client for RDP, Rdesktop is good, but is still considered beta code by some. RDP used to be very inefficient in comparison to ICA but the difference is now small if not negligible. Citrix has a number of features like load balancing which make it the better choice for large scale installations where the extra cost can be justified.

Both Citrix and Windows Terminal Server can introduce latency into the application if the servers are not sized correctly and the network is not sufficiently fast.

Tarantella (http://www.tarantella.com/) sits on a server between the desktop and the application servers. It aggregates output from Citrix on Windows and other applications running on UNIX and IBM mainframes, and sends the result to a browser on the desktop. It uses its own proprietary line protocol, AIP, which is apparently reasonable at low bandwidths. However, it increases latency because it sits between the user and the application and therefore slows down the connection between the two.

As mentioned above, CodeWeavers now produce a server version of their CrossOver Office product. This works by having the client securely connect to the central server and have an X session displayed back to it. This means that the communication to the central server is encrypted and compressed but also requires sufficient bandwidth to support it as it is based on X. No tests of the bandwidth requirement have been made but it is likely to be greater than for ICA (Citrix) or AIP (Tarantella). VNC is an OSS product developed by AT&T which is designed to display a user session running on another machine. It consists of a server and client which are both available for Windows , UNIX and GNU/Linux.

VNC allows applications to be run in one environment and the display to be run in another. It uses its own open protocol, RFB, over TCP/IP, which is not as efficient as either ICA (Citrix) or AIP (Tarantella) and so needs high network bandwidths (such as 100Mb/s) to work well. Unfortunately the Windows VNC server is also not as efficient as the UNIX version and can require more processing power than one would expect. VNC can be very useful for occasional systems administration use, allowing control of a desktop to be taken by a central person. In these circumstances, high latency could be acceptable.

5.4.4 Software Which Will Run Under An Emulator

If none of the above provides a way to run the application or a substitute then it may be possible to run it natively, but with its normal operating environment being emulated on top of an OSS operating system. A discussion of the issues relating to this approach can be found at:

http://www.linuxmednews.com/linuxmednews/967526746/index_html/

All of these techniques have license implications because they may involve running multiple copies of the proprietary application and/or operating system. This section is most likely to apply to Windows applications.

There are two types of emulation:

i. Hardware Emulation

Products like VMware and Win4lin provide hardware emulation. They allow a normal PC operating system to run as a user-level application by mimicking Intel PC hardware in software interfaces and thereby providing a virtual machine. This allows a legacy operating system and its applications to be run on top of an OSS platform.

VMware is not strictly an emulator - it allows most instructions to pass straight through to the processor, which means it will only run on an x86 architecture machine. It is the most complete offering but it is proprietary and can consume a lot of machine resources.

Win4lin is similar to VMware, and is also a proprietary product, but is less expensive. It can be a good solution in simple cases for example just running office applications. It is a component of the Lindows product which is being sold on low cost hardware to home users. (Because it apparently does not use unprivileged user accounts to maintain security, Lindows itself should not be recommended for Administrations without careful consideration of the security implications.)

Because the hardware emulation approach requires full licenses for the proprietary operating system and application, together with the cost of the emulator, it should be viewed as a way of running a small number of legacy applications which are difficult to migrate.

There are VMware and Win4lin server products which can reduce licensing costs if the proprietary software allows concurrent rather than potential user licensing.

There are OSS applications which will fully emulate an Intel IA-32 environment, for instance Bochs, but they are probably not ready for GoM use yet. -- Not production ready.

ii. Software Emulation

Software emulation allows programs written for a proprietary environment to run directly on the OSS operating system. Any system calls made by them are mapped onto the equivalent OSS system interface. This means that a copy of the proprietary operating system is no longer necessary.

Wine allows applications written for Windows to run on GNU/Linux by providing software emulation. The major problem Wine has to solve is the large number of Windows system calls (including bugs) that it has to support.

OSS Wine code is available from http://www.winehq.org/ or from CodeWeavers at: http://www.codeweavers.com/technology/wine/download.php/

CodeWeavers produce two proprietary products, CrossOver Office and CrossOver Plugin, which are based on Wine and designed to support specific Windows applications. Although the products are proprietary, code modifications are periodically contributed back to the OSS version of Wine.

CrossOver Office is designed to allow applications such as Office and Lotus Notes to run natively on GNU/Linux. There are some outstanding issues but the product is in active development. However, this approach may be appropriate for certain users depending on their requirements. CrossOver Office is now also available as a server product which means that it doesn't need to be fully installed on the desktop and can provide similar functionality to Citrix.

CrossOver Plugin is designed to allow browser plugins that normally only run on Windows to run in Netscape, Mozilla and Galeon on GNU/Linux. This product has been available for longer than CrossOver Office and works very well.

Using these techniques removes the cost of the Windows operating system license but not the application license. The application license needs to be scrutinized to make sure it doesn't forbid running the application without Windows. This restriction is used in some new Microsoft applications as a lock-in tactic, although legal enforcement is questionable.

5.4.5 Software Which Can Be Recompiled Under OSS

For applications written in-house or on behalf of the Government and for which source code is available, the software can be ported to run on an OSS platform. In general, the problem of porting source code in any language is not compilation but the code's usage of system libraries including both the graphical environment and operating system. This can mean a significant amount of manual intervention to migrate the code. In addition, any assumptions about the underlying environment, such as file naming, will make it necessary to either change the source code or replicate the environment, regardless of the language used.

I. Java

If Java software has been written according to the Java specification then the program should run without any problems. However, if any proprietary extensions have been used then the code will have to be changed to use standard modules instead.

II. Visual Basic

A proprietary product called DeLux (http://www.deluxsoftware.com/) can be used to convert Visual Basic code to Kylix (see item IV) and can be run under GNU/Linux natively. Netproject (http://www.netproject.com) have not been able to test this product. Microsoft's development tools can convert Visual Basic code to .NET and produce CIL code. The Mono OSS project allows this code to be run under GNU/Linux. Mono is currently being developed very rapidly and any given application may or may not work depending on the way it interacts with libraries such as the screen display.

III. C#

This is increasingly supported under GNU/Linux, and Novell (through its recently acquired Ximian) have produced a compiler as part of the Mono project, adding C# bindings to crucial components of the Gnome Desktop. The Mono project includes an interpreter that allows CIL code produced by proprietary development tools to be run on GNU/Linux unchanged. The Mono project and the use of the .NET development framework is a very lively area of OSS at the moment and the position changes very rapidly.

IV. Pascal and Delphi

Pascal as a free-standing language is little used these days, but it is the essential coding component of Borland's Delphi rapid development tool. Borland have a native GNU/Linux equivalent of Delphi that goes by the name of Kylix. Kylix 2 and Delphi 6 are stated to use compatible code syntax and have identical support environments.

V. C and C++

Programs written to ANSI standards should recompile and run as long as the underlying system libraries used are compatible. For instance programs written specifically for Windows will not in general compile and run correctly under GNU/Linux due to the very different set of calls to both the operating system and run-time libraries such as the windowing system. This mismatch can often be dealt with by compiling the code with Winelib, a part of the Wine project.

PART C

6 CONCLUSION

The Malaysian Government Interoperability Framework for OSS is constructed to be a “live” document, providing IT managers within the public sector examples, guidelines and recommendations for the adoption of, and migration to, OSS within their organisations. The constantly changing nature of technology and implementation methodologies mean that this document cannot possibly be exhaustive nor definitive.

However, it does serve as a useful reference in fulfilling the objectives of this document, which is primarily to ensure the interoperability and co-existence of new and legacy systems.

Throughout this document, the importance of open standards have been repeatedly emphasised, simply because the adherence to these standards, do more than any other measures that can be taken to ensure interoperability.

7 REFERENCES

i. The European Commission Interchange of Data between Administration (IDA) Open Source Migration Guidelines v1.0:

http://www.netproject.com/docs/migoss/v1.0

ii. Standards, Policies and Guidelines – Malaysian Government Interoperability Framework (MyGIF) v1.0, Malaysian Administrative Modernisation and Management Planning Unit, August 2003:

http://www.mampu.gov.my/mampu/bi/program/ict/ISPlan/ispdoc/Interoperability%20Framework.pdf

iii. The United Kingdom e-Government Interoperability Framework v6.0, The Cabinet Office, e-Government Unit, Technology Policy Team, Interoperability Policy Advisor, April 2004:

http://www.govtalk.gov.uk/schemasstandards/egif.asp

8 ABBREVIATIONS AND ACRONYMS

The abbreviations and acronyms throughout this document are similar to the ones detailed in MyGIF, with some expansion as given below:

Abbreviation/Acronym	Expension
Multimedia	
Codec	Short *encoder/decoder*, this term is used to describe the process to encoding or decoding digital data (usually video, images or audio) from one form to another. For example, digital images are *encoded* into the JPEG format for storage and *decoded* from the format for display onto the screen. Encoding involves compressing the data and decoding, decompressing.
Ogg Vorbis	It is an audio compression format which is patent free and open source. It was started after litigation possibilities were raised in using the popular MP3 format which had patents held by the Fraunhofer Institute. Ogg is the container format and Vorbis is the compression scheme.
MP3	MPEG-1 Audio Layer-3
Linux	
RPM	RPM Package Manager (formerly known as Redhat Package Manager) is a package management system, primarily for Linux which installs, updates, uninstalls, verifies and queries software. It is used by many Linux distributions for this very purpose.
DPKG	DPKG is the base of the Debian Package Management System which is similar to RPM. In fact, RPM was originally based on DPKG, though simplified.
APT	Advanced Packaging Tool – a high-level package management system which manages the retrieval, configuration and installation of software packages. It is used mainly for DPKG-based packages but has been expanded to handle RPM-based ones as well.
TGZ	File extension which means that a collection of files within a software package have been inserted into a Tar archive and compressed with gzip – the TGZ files are commonly referred to as gzipped tarballs. It is also the package management system used in some Linux distributions. As a package management system, though, it is not as advanced as others.
emerge	A package management system used by primarily by the Gentoo Linux distribution. It works similarly like APT, in that is obtains the source code for a package and then configures and installs it.
tar	An archive program. Tar stands for "tape archive".

Table 8.1: Abbreviations & Acronyms

OSCC
OPEN SOURCE COMPETENCY CENTRE
MALAYSIA

MALAYSIAN PUBLIC SECTOR OPEN SOURCE COMPETENCY CENTRE
3rd Floor, APT E302-E304
Enterprise Building 3, Persiaran APEC
63000 Cyberjaya, Selangor.
Phone: 03 83191200 Fax: 03 83193206
Email: contact_centre@oscc.org.my
Website: http://opensource.mampu.gov.my

MAMPU

PRIME MINISTER DEPARTMENT
Level 6, Block B2
Federal Government Administration Centre
62502 Putrajaya, Malaysia