

**MALAYSIAN PUBLIC SECTOR
OPEN SOURCE SOFTWARE (OSS)
PROGRAM**

**OPEN SOURCE SOFTWARE
REFERENCE ARCHITECTURE
(OSSRA)**

COPYRIGHT

- The Government of Malaysia retains the copyright of this document.

Table Of Content

1 Purpose of the document.....	1
1.1 Supplementary Documents.....	1
1.2 Objectives	2
1.3 Scope.....	2
1.4 Audience.....	3
2 OSS Enterprise Architecture.....	4
3 Network Architecture.....	7
3.1 Domain Name System (DNS).....	10
3.2 Dynamic Host Configuration Protocol (DHCP).....	10
3.3 Proxy Server.....	10
3.4 Wireless.....	11
3.5 Mobile Messaging Server.....	11
3.6 Voice over Internet Protocol (VoIP).....	11
3.7 Network Security Layer	11
3.7.1 Firewall.....	11
3.7.2 Intrusion Detection System (IDS).....	12
3.7.3 Virtual Private Network (VPN).....	12
3.7.4 Authentication Server.....	12
3.8 Network Management Layer.....	13
3.8.1 Simple Network Management Protocol (SNMP).....	13
4 Server Architecture.....	14
4.1 Web Server.....	16
4.2 Mail Transfer Agent (MTA).....	16
4.3 Mail Delivery Agent (MDA).....	16
4.4 Virtualization.....	16
4.5 Directory Server.....	17
4.6 Database Server.....	17
4.7 Server Security Layer.....	18
4.7.1 Host-Based Intrusion System (HIDS).....	18
4.7.2 Backup Server.....	18

4.8 Server Management Layer.....	18
4.8.1 Simple Network Management Protocol (SNMP).....	19
5 Database Architecture.....	20
5.1 Database Server.....	22
5.2 Open Database Connectivity (ODBC).....	22
6 APPLICATION ARCHITECTURE.....	23
7 Client Architecture.....	26
a) Fat Client.....	26
b) Thin Client.....	27
c) Virtualized Desktop.....	28
8 Conclusion.....	30
REFERENCES	31

1 Purpose of the document

The Open Source Software Reference Architecture (OSSRA) document is designed to assist and guide the System Administrator and Technical Support within government agencies in architecting and designing the ICT infrastructure in compliance with all current published ICT policies, guidelines and standards.

OSSRA was developed based on the general principles as stipulated in the following references :

1. Arahan Teknologi Maklumat MAMPU, Disember 2007. Akta Aktiviti Kerajaan Elektronik 2007
2. Modernisation and Management Planning Unit, MAMPU, 15 January 2002. "The Malaysian Public Sector Management of ICT Security Handbook (MyMIS) Version 2.0"
3. Inisiatif Melindungi Aset ICT Sektor Awam. MAMPU, November 2006
4. Rangka Dasar Keselamatan Teknologi Maklumat dan Komunikasi Kerajaan, Pekeliling Am Bil. 3 Tahun 2000, Jabatan Perdana Menteri Malaysia

Apart from the above references, this document was also developed based on other references as stated in Appendix A.

1.1 Supplementary Documents

This document should be read together with **The Malaysian Government Interoperability Framework for Open Source software (MyGIFOSS)** and **Open Source Software (OSS) Implementation Guidelines**. This document should also be read concurrently with The Malaysian Public Sector OSS Master Plan and Web Application Guidelines.

1.2 Objectives

The objectives of the Open Source Software Reference Architecture are :

- To ease government agencies in implementing ICT solutions using OSS technology.
- To standardize OSS technology deployment in Public Sector.

1.3 Scope

The scope of the OSSRA is based on an OSS Enterprise Architecture that covers the following components :

- Network Architecture
- Server Architecture
- Database Architecture
- Application Architecture
- Desktop Applications

The foundation for the OSSRA components is a system running on Open Source Operating System like Linux, FreeBSD and OpenSolaris. Each OSSRA components provides a discrete function. Agencies can deploy the subset of OSSRA components which meet the requirements specific to their ICT environment.

The above scope will be supported by Management Software and Security solutions and will be guide by MAMPU policies and procedures.

1.4 Audience

This document is intended for use by Public Sector Agencies to gain understanding on how to implement OSS in their respective agencies. The target audiences for this document are the Chief Information Offices (CIO's) ,Head of Departments (HODs), IT Managers and all ICT personnel.

2 OSS Enterprise Architecture

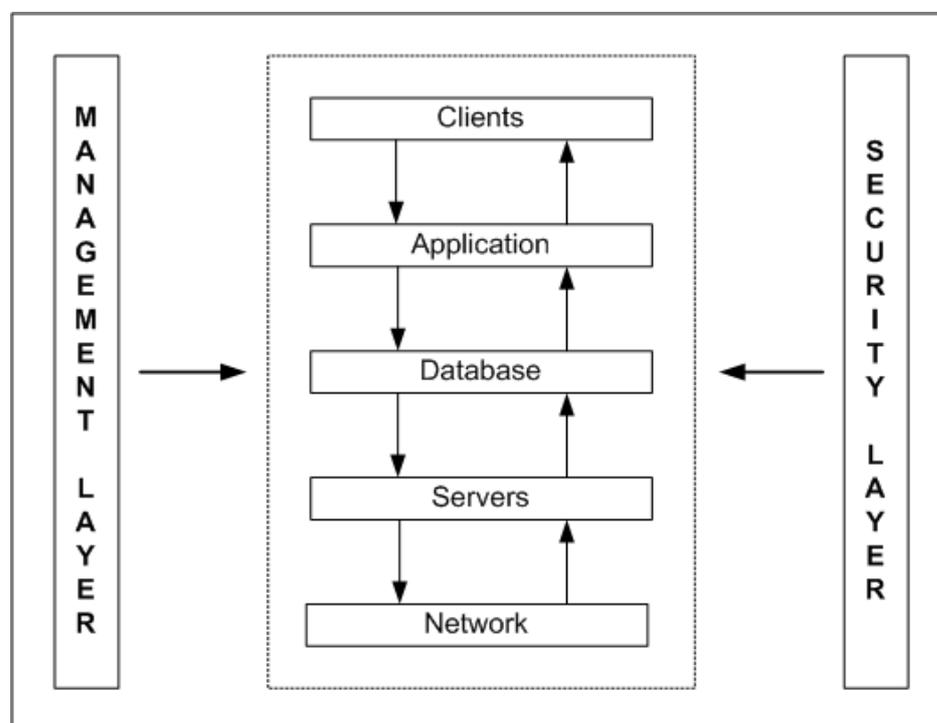


Figure 1 : OSS Enterprise Architecture

OSS Enterprise Architecture is the description of the current and/or future structure and behavior of an organization's processes aligned with the organization's core goals and strategic direction in deploying OSS technologies. The architecture in Figure 1 addresses documenting and understanding the discrete enterprise structural components, typically within the following **five (5)** components :

- **Clients**

A client is an application or system that accesses a service on another computer system known as a server by way of a network.

- **Application**

Application architecture is a software engine that delivers application and web services to clients. It handles the business

logic and data access of the application.

- **Database**

Database architecture is a computer program that provides database services to other computer programs or computers, as defined by the client-server model.

- **Servers**

Server architecture is a collection of platforms and infrastructure that performs services to the connected clients. Servers are devices designed to run applications for extended period of time with minimal human direction.

- **Network**

Network architecture is an interconnection of group of computers and computer devices that perform according to basic reference model. It often classified according to the network topology upon which the network is based.

All the components are supported by the following **two (2)** pillars:

- **Management Layer**

Management tools pertain the operation, administration, maintenance, and provisioning of networked systems.

- Operation keep the network up and running smoothly. It also monitors the network to spot problems.
- Administration keep track of resources in the network and how they are assigned. It includes all the "housekeeping" that is necessary to keep the network under control.
- Maintenance is concerned with performing repairs and upgrades.
- Provisioning is concerned with configuring resources in the network to support a given service.

- **Security Layer**

Security tools pertain sets of functions that protects telecommunications networks and systems from unauthorized access by persons, acts, or influences and that includes many functions, such as creating, deleting, and controlling security services and mechanisms.

Why do we need OSS Enterprise Architecture ?

The OSS Enterprise Architecture is important for the following reasons :

- To standardize the structure of OSS deployment in agencies.
- To perform a standard way of OSS infrastructure management and OSS security enforcement in an ICT environment.
- To minimize the OSS environment adoption learning curve among government officers when they are transferred to different agencies.

3 Network Architecture

OSSRA Network Architecture defines the standard and technologies to support the other components in Enterprise Architecture. It provides the foundation and base solutions for agencies to build ICT environments related to network infrastructure using OSS.

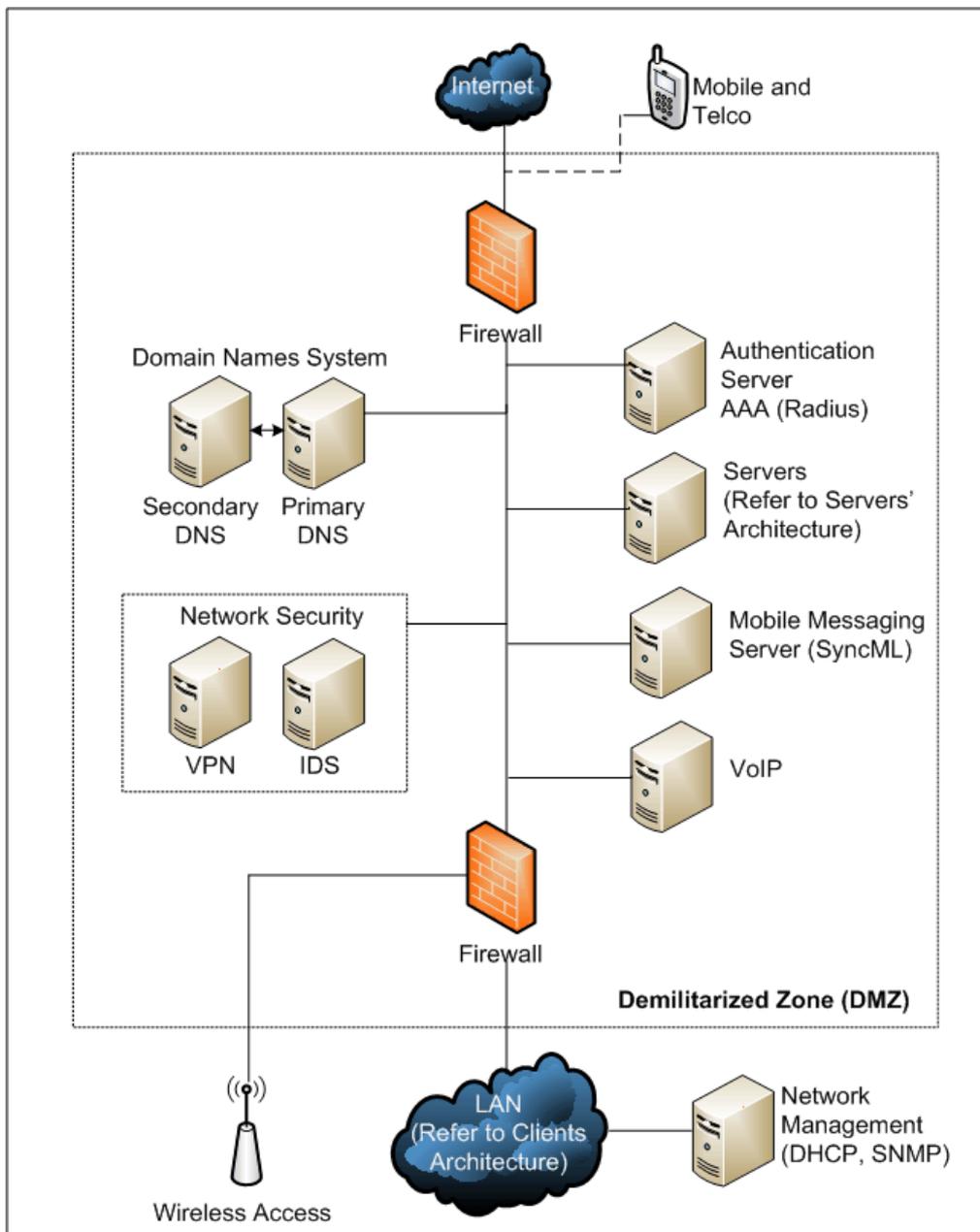


Figure 2 : Network Architecture

The Network Architecture in Figure 2 defines the end to end management of the communications session and includes the access and delivery protocols. It's combination of **Supporting Network Services** like Simple Mail Transfer Protocol (SMTP), Internet Message Access Protocol (IMAP), Post Office Protocol version 3 (POP3), Multipurpose Internet Mail Extensions (MIME) and Simple Network Management Protocol (SNMP). This combination also include **Service Transport** like Transmission Control Protocol (TCP), Internet Protocol (IP), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) and Wireless Application Protocol (WAP).

Building a Network Architecture with security in mind is important for government agencies. Several security guidelines and procedures have been developed by MAMPU and distributed to agencies to assist in the compliance of security policies. OSSRA heavily refers to MyMIS , chapter 4.4 Network System (page 4-13) and Arahan Teknologi Maklumat, Disember 2007 Bab III Keperluan- Keperluan Keselamatan ICT (page 17).

Firewall is an important tool in securing the Network Architecture but is not a total solution to address Network Security concerns. A number of solutions need to be installed properly and proper use of firewall zoning need to implemented. Network traffic from Internet must be separated and default access policy in the firewall should be set to "DENY".

Access to Internet from Local Area Network (LAN) must be done through proxy and web filtering mechanism. All computers and devices in the LAN area must not be allowed directly to access to the Internet unless it is approved by management.

Any server that need to access the Internet should be in the Demilitarized Zone (DMZ) of the Firewall. Separate IP's segment from

LAN should be used and default firewall rules that deny all will limit the access from either LAN or the Internet.

Intrusion Detection System (IDS) like Network IDS (NIDS) and Host IDS (HIDS) should be implemented and reports and logs reviewed regularly. Early detection will help to identify any risk that will impact the Network and the servers.

Any remote access from external sites to the network should be done using Virtual Private Network (VPN) technology. By using VPN, network connections can be protected from spoofing attempts and data can be securely transferred between servers and remote access sites. All VPN access must be controlled and monitored effectively by using different IP network address segments and firewall zoning.

Domain Name System (DNS) should be duplicated across two servers. The two servers should be separated into two different locations for redundancy purposes. This will make sure any DNS server that is malfunctioning due to hardware or network problems will not affect DNS service as the other DNS server will take over the function.

Dynamic Host configuration Protocol (DHCP), should be used by network administrators to easily allocate and manage addresses the network in the LAN. DHCP servers should be placed in the LAN area only to respond to DHCP requests from client PCs or notebooks. No DHCP server should be in the DMZ area for security reasons. Static IP address should be enabled for devices in the DMZ area.

By implementing Simple Network Management Protocol (SNMP), it will help the network administrator to effectively monitor the network devices. Most of Network Monitoring Software (NMS) are using SNMP as the way to monitor and manage all devices in the network.

3.1 Domain Name System (DNS)

The Domain Name System (DNS) associates various sorts of information with so-called domain names; most importantly, it serves as the "phone book" for the Internet by translating human-readable computer hostnames, e.g. www.example.com, into the IP addresses, e.g. 208.77.188.166, that networking equipment needs to deliver information. Below are some OSS based DNS that can be deployed;

- Bind DNS (<http://www.isc.org/products/BIND/>)
- TinyDNS (<http://www.tinydns.org>)

3.2 Dynamic Host Configuration Protocol (DHCP)

Dynamic Host Configuration Protocol (DHCP) is a protocol used by networked devices to obtain various parameters necessary for the clients to operate in an Internet Protocol (IP) network. By using this protocol, system administration workload greatly decreases, and devices can be added to the network with minimal configuration.

- ISC DHCP (<http://www.isc.org>)

3.3 Proxy Server

Proxy server is a server which services the requests of its clients by forwarding requests to other servers. A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource, available from a different server. The proxy server provides the resource by connecting to the specified server and requesting the service on behalf of the client. Below is the OSS based proxy server that can be deployed;

- Squid (<http://www.squid-cache.org>)

3.4 Wireless

The term wireless is normally used to refer to any type of electrical or electronic operation which is accomplished without the use of a "hard wired" connection. Wireless communication is the transfer of information over a distance without the use of electrical conductors or "wires".

- FreeRadius (<http://www.freeradius.org>)
- Coova Chilli (<http://www.coova.org>)

3.5 Mobile Messaging Server

Mobile Messaging Server provides push email, address book and calendar (PIM) data synchronization, and device management for wireless devices, leveraging standard protocols such as SyncML. For users, this means BlackBerry-like capabilities on commodity handsets.

- Funambol (<http://www.funambol.com>)

3.6 Voice over Internet Protocol (VoIP)

Voice over Internet Protocol is a protocol used for voice transmission through the internet. This technology allow users to make voice calls using broadband internet connections instead of regular phone line.

- Asterisk (<http://www.asterisk.org>)

3.7 Network Security Layer

Security layer in network architecture plays an pivotal role in protecting the network and the network-accessible resources from unauthorized access and vulnerabilities. Below are some OSS based solutions categorized by its functions.

3.7.1 Firewall

A firewall is a dedicated appliance, or software running on another

computer, which inspects network traffic passing through it, and denies or permits passage based on a set of rules. Below are some OSS based firewalls that can be deployed;

- Linux iptables (<http://coombs.anu.edu.au/~avalon/>)
- IPFW (<http://www.freebsd.org/doc/handbook/firewalls-ipf.html>)
- Monowall (<http://m0n0.ch/wall/>)
- Smoothwall (<http://www.smoothwall.org>)

3.7.2 Intrusion Detection System (IDS)

An intrusion detection system is used to detect several types of malicious behaviors that can compromise the security and trust of a computer system. This includes network attacks against vulnerable services, data driven attacks on applications, host based attacks such as privilege escalation, unauthorized logins and access to sensitive files, and malware (viruses, trojan horses, and worms). Below is the OSS based IDS that can be deployed;

- Snort (<http://www.snort.org>)

3.7.3 Virtual Private Network (VPN)

A virtual private network (VPN) is a communications network tunneled through another network, and dedicated for a specific network. Usually, a VPN is set up using the internet as the bridge. Below are some OSS based VPN that can be deployed;

- OpenVPN (<http://www.openvpn.net>)
- FreeS/WAN (<http://www.freeswan.org>)

3.7.4 Authentication Server

Authentication servers are servers that provide authentication services to users or other systems. Users and other servers authenticate to

such a server, and receive cryptographic tickets. These tickets are then exchanged with one another to verify identity. Below are some OSS based authentication server that can be deployed;

- Radius (<http://www.freeradius.org>)
- Tacacs+ (<http://www.gazi.edu.tr/tacacs/>)

3.8 Network Management Layer

Network management layer refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems. Below are some OSS based solution categorized by its function.

3.8.1 Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) forms part of the internet protocol suite as defined by the Internet Engineering Task Force (IETF). SNMP is used in network management systems to monitor network-attached devices for conditions that warrant administrative attention. It consists of a set of standards for network management, including an Application Layer protocol, a database schema, and a set of data objects. Below are some OSS based network management software that can be deployed;

- Nagios (<http://www.nagios.org>)
- Zenoss (<http://www.zenoss.com>)

4 Server Architecture

The Server Architecture defines the Server technologies that help agencies in building an ICT environment using Open Source software's and solutions.

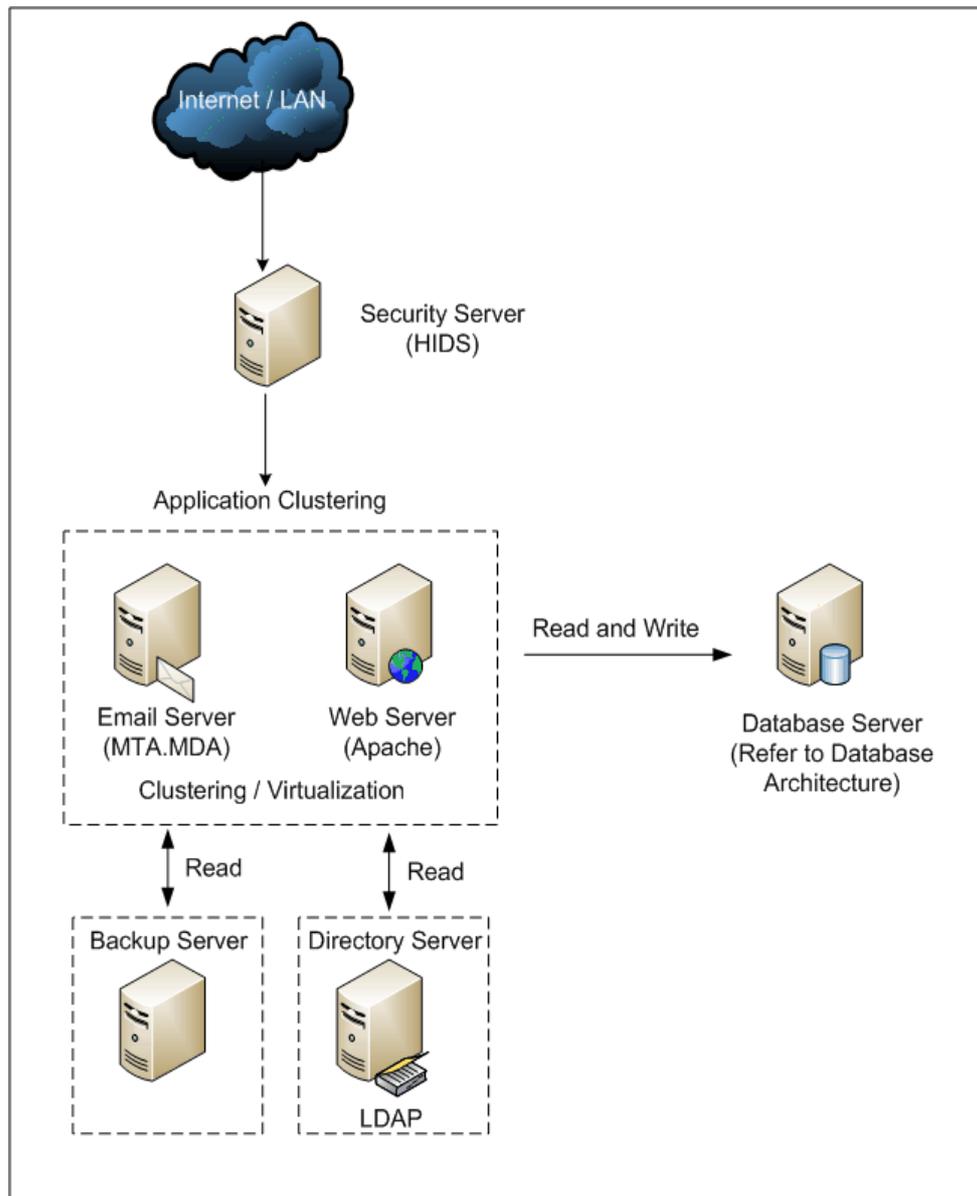


Figure 3 : Server Architecture

The Server Architecture in Figure 3 outlines the collection of platforms and infrastructure standards enabling component based architecture. This include Support Platforms like Operating System and hardware. This combination also include Delivery servers like Web Servers, Media Servers and Portal Servers.

Backup and redundancy are important in the Server Architecture. Hardware clustering and using virtualization to consolidate hardware are the approaches that should be taken by the agencies. Clustering technology can be used for solutions where high performance, load balancing and fault tolerance are required.

Virtualization can be used for hardware consolidation that will reduce the number of physical hardware used. With high performance hardware that are becoming cheaper, a single hardware can run several services like email, web server, application server depending on the hardware specification.

Servers Architecture services should be protected by firewall by zoning it in a DMZ zone or even to the Local Area Network (LAN) zone. Host Intrusion Detection System (HIDS) should be installed in each of the servers to help detect any server security breach.

Backup server will typically host a tape media drive and installed with backup software. At the same time, a backup server by itself can also be the cold standby server which can act as a temporary server.

User management is part of a system administrator's job function. A directory server centralizes users information, applications settings, group data, policies and access control information into centralize network-based repository. It's much more easier to start centralize users management at the initial stage of build an ICT environment.

4.1 Web Server

A computer program that is responsible for accepting HTTP requests from clients, which are known as web browsers, and serving them HTTP responses along with optional data contents, which usually are web pages such as HTML documents and linked objects. Below are some OSS based Web Server that can be deployed:

- Apache (<http://www.apache.org>)
- Lighttpd (<http://www.freshmeat.net/projects/lighttpd>)

4.2 Mail Transfer Agent (MTA)

A mail transfer agent (MTA), is a computer program or software agent that transfers electronic mail messages from one computer to another. Below are some OSS based MTA that can be deployed:

- Postfix (<http://www.postfix.org>)
- Sendmail (<http://www.sendmail.org>)
- Qmail (<http://www.qmail.org>)
- Exim (<http://www.exim.org>)

4.3 Mail Delivery Agent (MDA)

A Mail Delivery Agent (MDA) is software that delivers e-mail messages right after they've been accepted on a server, distributing them to recipients' individual mailboxes or forwarding to another SMTP server . Below are some OSS based MDA that can be deployed:

- DBMail (<http://www.dbmail.org>)
- Dovecot (<http://www.dovecot.org>)

4.4 Virtualization

Virtualization is a technique for hiding the physical characteristics of computing resources from the way in which other systems, applications, or end users interact with those resources. This includes making a single physical resource appear to function as multiple logical resources; or it can include making multiple physical resources appear as a single logical resource. Below are some OSS based Virtualization that can be deployed:

- XEN (<http://www.xensource.com>)
- VirtualBox (<http://www.virtualbox.org>)

4.5 Directory Server

Directory Server stores and organizes information about a computer network's users and network resources, and that allow network administrators to manage users' access to the resources. Additionally, directory services act as an abstraction layer between users and shared resources. Below are the OSS based directory server than can be deployed.

- LDAP (<http://openldap.org>)

4.6 Database Server

A database server is a computer program that provides database services to other computer programs or computers, as defined by the client-server model. Below are some OSS based Database Server that can be deployed:

- MySql (<http://www.mysql.com>)
- PostgreSQL (<http://www.postgresql.org>)
- Firebird (<http://www.firebirdsql.org>)

4.7 Server Security Layer

Security layer in server architecture plays an pivotal role in protecting the server and the server-accessible resources from unauthorized access and vulnerabilities. Below are some OSS based solutions categorized by its functions.

4.7.1 Host-Based Intrusion System (HIDS)

A host-based intrusion detection system (HIDS) is an intrusion detection system that monitors and analyzes the internals of a computing system rather than on its external interfaces. Host-Based IDS's monitor all or parts of the dynamic behavior and of the state of a computer system. Below is OSS based HIDS that can be deployed:

- OSSEC (<http://www.ossec.net>)

4.7.2 Backup Server

Backup server is a backup system that allows the administrator to set up a single master backup server to back up multiple hosts over network to tape drives/changers or disks or optical media. Below are some OSS based Backup Server that can be deployed:

- Amanda (<http://www.amanda.com>)
- Bacula (<http://www.bacula.org>)

4.8 Server Management Layer

Network management layer refers to the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems. Below are some OSS based solution categorized by its function.

4.8.1 Simple Network Management Protocol (SNMP)

The Simple Network Management Protocol (SNMP) forms part of the internet protocol suite as defined by the Internet Engineering Task Force (IETF). SNMP is used in network management systems to monitor network-attached devices for conditions that warrant administrative attention. It consists of a set of standards for network management, including an Application Layer protocol, a database schema, and a set of data objects. Below are some OSS based network management software that can be deployed;

- Nagios (<http://www.nagios.org>)
- Zenoss (<http://www.zenoss.com>)

5 Database Architecture

The Database Architecture define database technologies that can help agencies to build an ICT environment using Open Source software's and solutions.

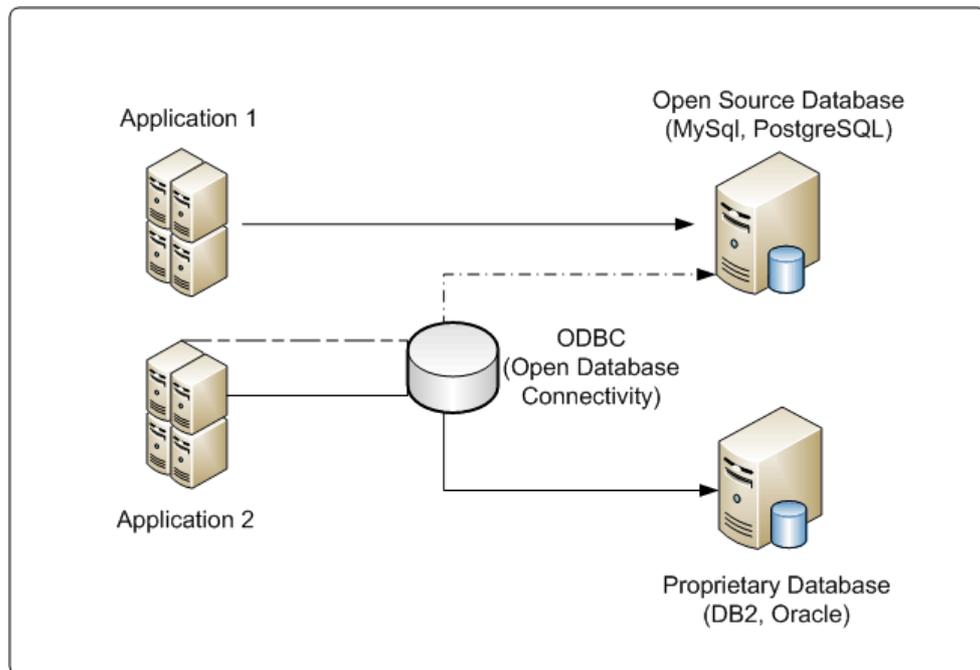


Figure 4 : Database Architecture

The Database Architecture in Figure 4 refer to collection of programs that enable storage, modification and retrieval of information from a database, and various techniques and devices for storing large amounts of data. A database management system (DBMS) is a software application that provides management, administration, performance and analysis tools for databases. The design of the database enables a computer program to quickly select desired pieces of data.

The right hardware configuration is essential for making sure the database achieves acceptable performance and reliability. Redundant Arrays of Independent Disks (RAID) hardware or software should be used. RAID 10 and RAID 5 are the recommended options for database servers, depending on type of database operations (read-write or read-only). Memory is also essential aspect in database performance. The larger the memory , the better the performance of the database will be.

With an existing database, it is helpful to map out the necessary read/write dependencies and how specific data will be accessed by the web application. If a database contains a number of tables that parts of the application need to access, or it stores collections of unrelated types of data, partitioning the data into appropriate tables allows you to move them to discrete database servers, where they can be scaled individually when the data volume become huge.

Database abstraction layer unifies the communication between the web application and databases by providing a single programming interface to the application developer and hiding the database specifics behind this interface as much as possible.

By adopting database abstraction layer, the web application will not be locked-in to specific database system, and changing the database

behind the application would no longer require massive changes to the source code of the application to adopt to the new database system. This increases the portability, interoperability and scalability of the web application, since the same piece of application can work with multiple database systems.

Examples of database abstraction layer include OpenDBX, PEAR, ADOdb, JDBC, ODBC, OLE-DB, etc. Some web application frameworks provide database abstraction layer as one of their component.

5.1 Database Server

A database server is a computer program that provides database services to other computer programs or computers, as defined by the client-server model. The term may also refer to a computer dedicated to running such a program. Below are some OSS based Database Server that can be deployed:

- MySQL (<http://www.mysql.com>)
- PostgreSQL (<http://www.postgresql.org>)
- Firebird (<http://www.firebirdsql.org>)

5.2 Open Database Connectivity (ODBC)

Open Database Connectivity (ODBC) provides a standard software API method for using database management systems (DBMS). The designers of ODBC aimed to make it independent of programming languages, database systems, and operating systems. Below are some OSS based ODBC that can be deployed:

- iODBC (<http://www.iodbc.org>)
- UnixODBC (<http://www.unixodbc.org>)

6 APPLICATION ARCHITECTURE

The Application Architecture define application component that can help agencies to build an ICT environment using Open Source software's and solutions.

The Application Architecture delivers applications and web services to clients, typically through the Internet and using the HyperText Transfer Protocol. Application servers are distinguished from web servers by the extensive use of server-side dynamic content and frequent integration with database engines.

An application server handles most, if not all, of the business logic and data access of the application. The main benefit of an application server is the ease of application development, since applications need not be programmed; instead, they are assembled from building blocks provided by the application server.

Application servers typically bundle middleware to enable applications to intercommunicate with dependent applications, like web servers, database management systems, and chart programs. Some application servers also provide an Application Programming Interface (API), making them operating system independent. Portals are a common application server mechanism by which a single point of entry is provided to multiple applications.

Programming is minimized because the application server has the user interface instructions already built in. The instructions are contained in output objects, and database datatypes are preassigned to output objects. Configuring the application means the developer is assigning elements of the application to database datatypes. When the server is running, data is requested by the client, causing the assigned user

interface instructions to be sent to the client along with the data. Client-side data integrity is refined by programming hook functions, which are simultaneously sent to the client.

Application servers are built on programming language that users may not really know about. Examples of Application servers are JBoss Enterprise Application Platform, Apache Geronimo and SUN Java System Application Server.

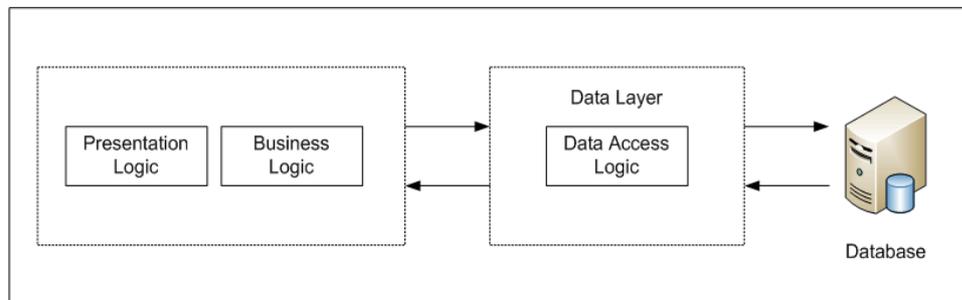


Figure 5 : 2-Tier Application Architecture

In the early days of web computing, most websites deployed a 2-tier architecture as in Figure 5, which consisted of a web server that processed HTTP requests and a database server that provided a back-end data store. Application logic that served the website resided on the web server, which interacted directly with databases and generated dynamic web pages based on the query results.

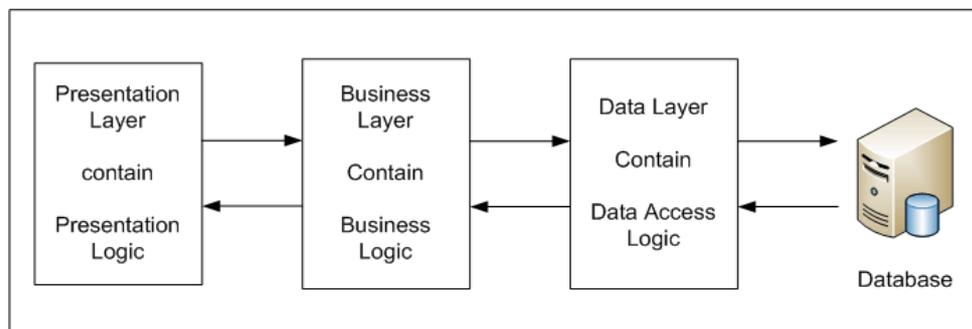


Figure 6 : 3-Tier Application Architecture

Separating the input/output-bound web operations (such as displaying

static content) from processor-bound activities (such as performing user transactions) by deploying a 3-tier architecture as in Figure 6 effectively isolates and resolves scalability and maintainability bottlenecks in both of the 2-tier scenarios described above. The 3-tier architecture model adds an application server tier to handle the business logic of a web application. With a 3-tier architecture, adding more web server tier machines can address the problem of slow static web page response times. If response times for processing transaction requests are slow, adding more application-server tier machines can improve their performance.

7 Client Architecture

The client device is used as the front-end delivery channel for the enterprise architecture. The client architecture can consist of either one of the following components:

- a) Fat client
- b) Thin Client
- c) Virtualized desktop

a) Fat Client

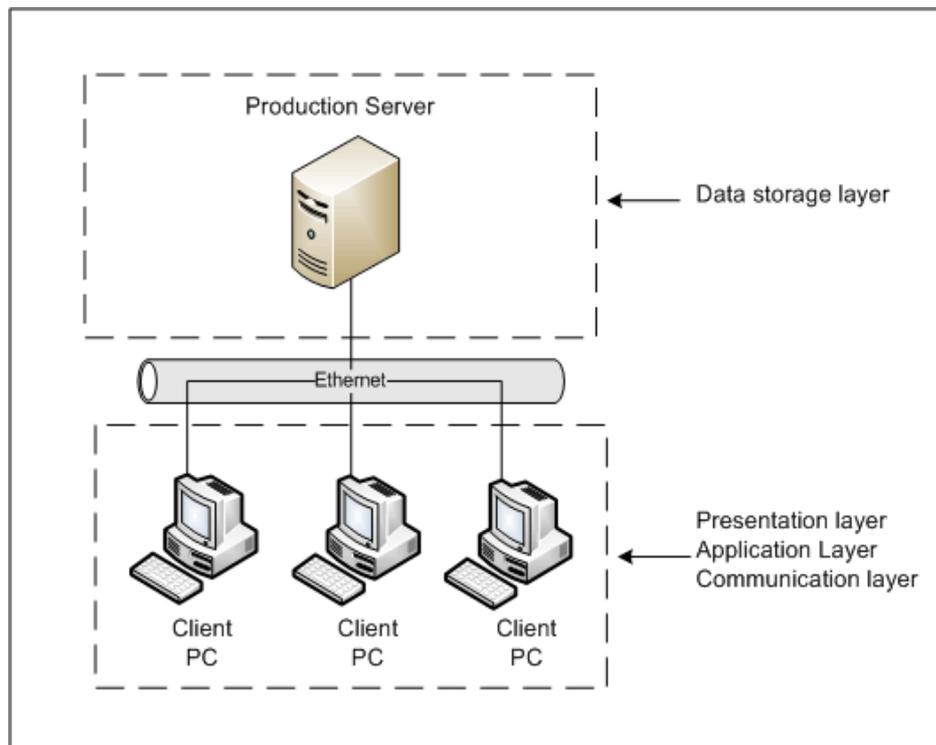


Figure 7 : Fat Client

A fat client as in Figure 7 is a networked computer with most resources installed locally rather than distributed over a network. However, a fat client still requires at least periodic connection to a network to enable access to web servers, file sharing and other processes that need to be

done by a server.

Most PCs (personal computer) are fat client because they have their own hard drives, CD/DVD drives, software application and so on. Fat clients are almost unanimously preferred by network users because they are very customizable and the user has more control over what programs are installed and specific system configuration.

b) Thin Client

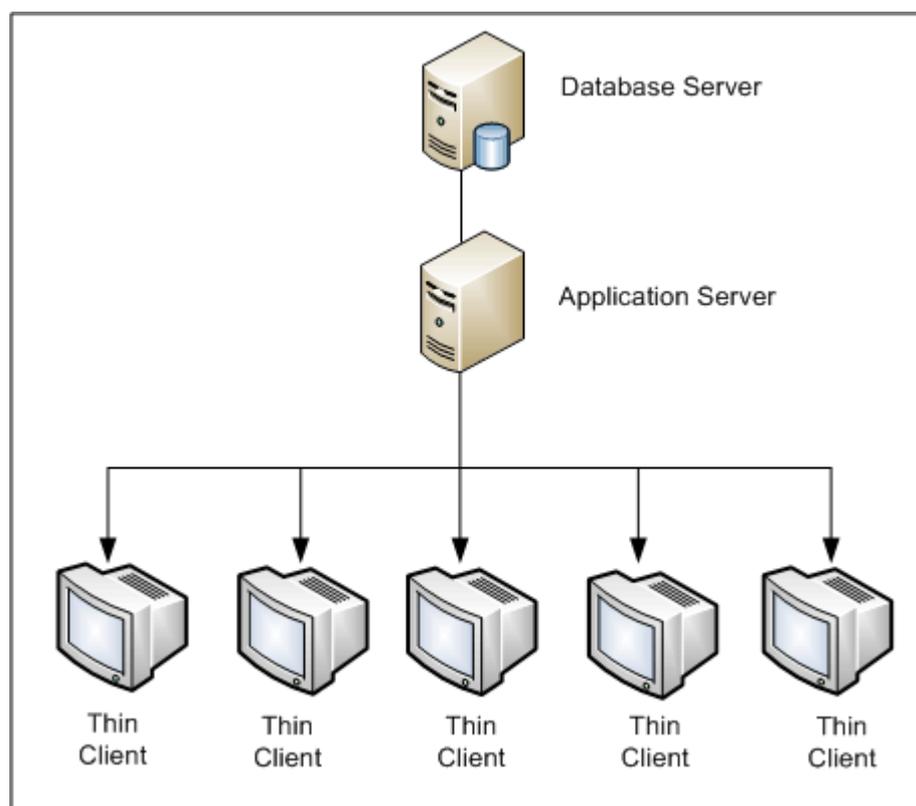


Figure 8 : Thin Client

A thin client as in Figure 8 is a client computer in client-server architecture networks which depends primarily on the central server for processing activities, and mainly focuses on conveying input and output between the user and the remote server.

Many thin client devices run only web browsers or remote desktop

software, meaning that all significant processing occurs on the server. However, recent devices marketed as thin clients can run complete operating systems such as Debian GNU/Linux, qualifying them as diskless nodes or hybrid clients.

c) Virtualized Desktop

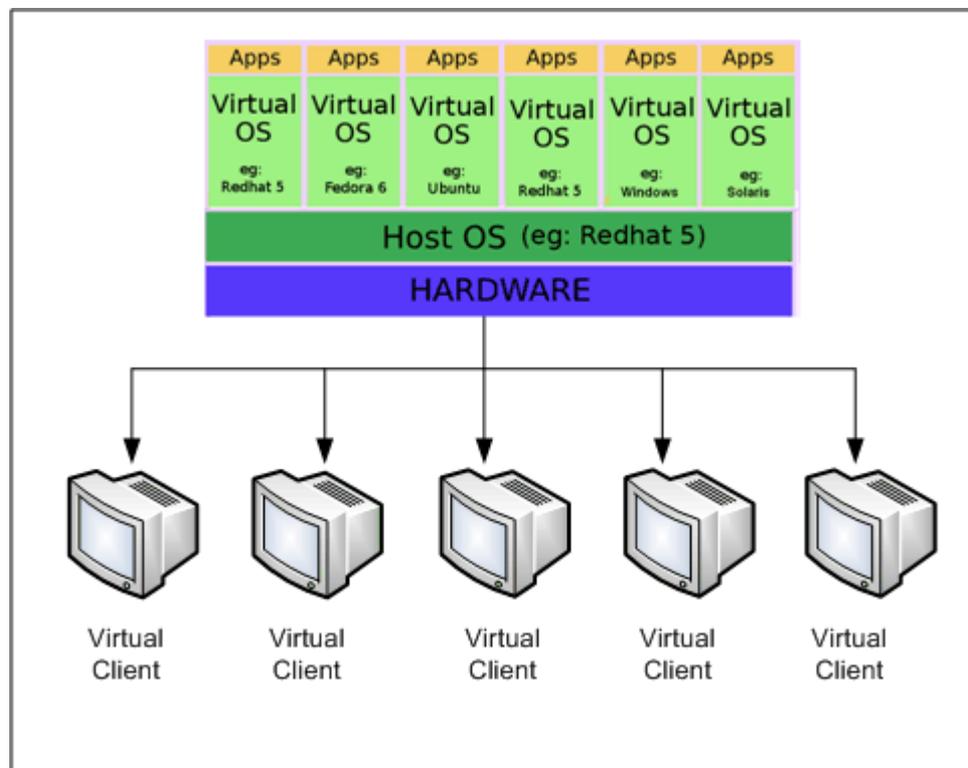


Figure 9 : Virtualized Desktop

There are four (4) distinct classes of desktop virtualization which are Single Remote Desktop, Shared Desktops, Virtual Machine (VM) Desktops and Physical PC Blade Desktops. A virtualized desktop as in Figure 1.9 can be contrasted with a traditional local PC Desktop, where the user directly accesses the desktop operating system and all of its peripherals physically. Differ from that, when a desktop is virtualized, its keyboard, mouse and video display are typically redirected across a network via a desktop remoting protocol such as Remote Desktop Protocol (RDP), Independent Computing Architecture (ICA) and Virtual

Network Computing (VNC).

Once a desktop is virtualized, it becomes accessible over any suitable network connection, on any device with similar characteristics. The most common case is a user simply needing direct access to a single PC machine's desktop from a remote location over a network. When many user desktops need to be hosted and managed centrally, that's where the Shared, VM and Physical Desktop models are used. Running user desktops (and applications) centrally provides significant value and benefits over the traditional local PC model, including improved security by keeping desktops in secure data centers, lowering management costs through centralization, and the ability to effectively share PC compute power across many users.

8 Conclusion

The Open Source Software Reference Architecture (OSSRA) is constructed to be “live” document, providing IT personnel within the public sector, guidelines and recommendations for adoption OSS within their organizations.

This document serve as a useful reference in fulfilling the objectives of this document, which is primarily to ensure government agencies to build and deploy open source enterprise wide solutions in their ICT environment.

APPENDIX A

REFERENCES

1. Modernisation and Management Planning Unit, MAMPU, December 2007. "Arahan Teknologi Maklumat". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
2. Modernisation and Management Planning Unit, MAMPU, 20 October 2006. "Arahan KSN – Langkah-Langkah Untuk Keselamatan Rangkaian Setempat Tanpa Wayar (Wireless Local Area Network) di Agensi-Agensi Kerajaan". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
3. Modernisation and Management Planning Unit, MAMPU, 1 October 2006. "Pekeliling Am Bilangan 3 Tahun 2000 – Rangka Dasar Keselamatan Teknologi Maklumat dan Komunikasi Kerajaan". Kuala Lumpur : Percetakan Nasional Malaysia Berhad
4. Modernisation and Management Planning Unit, MAMPU, 2006. "Open Source Software (OSS) Policy Handbook". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
5. Modernisation and Management Planning Unit, MAMPU, April 2006. "OSS Implementation Guideline". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
6. Modernisation and Management Planning Unit, MAMPU, February 2006. "Malaysian Government Interoperability Framework for Open Source Software (MyGIFOSS) ". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
7. Modernisation and Management Planning Unit, MAMPU, January 2005. "Malaysian Public Sector Open Source Software (OSS) Master Plan". Kuala Lumpur : Percetakan Nasional Malaysia Berhad.
8. Modernisation and Management Planning Unit, MAMPU, August 2003. "Malaysian Government Interoperability Framework (MyGIF) Version 1.0". Kuala

Lumpur : Percetakan Nasional Malaysia Berhad.

9. Modernisation and Management Planning Unit, MAMPU, 15 January 2002.
“The Malaysian Public Sector Management of ICT Security Handbook (MyMIS) Version 2.0”. Kuala Lumpur : Percetakan Nasional Malaysia Berhad.