



**Office of the Prime Minister
Central Information Management Unit
Technical Report**

CIMU T 0001:2002

e-Government Interoperability Framework

Version: 1.0

Effective Date: 3rd July 2002

Table of Contents

TABLE OF CONTENTS	2
FIGURES.....	3
TABLES.....	3
1. PURPOSE.....	4
2. WHO SHOULD READ THIS REPORT	4
3. SCOPE OF APPLICABILITY	4
4. DEFINITIONS.....	5
5. SECURITY CLASSIFICATION.....	5
6. EXECUTIVE SUMMARY	6
7. OVERVIEW.....	7
8. INTERCONNECTIVITY	8
8.1 SECURITY	8
8.1.1 Citizen Identification, Authentication and Authorisation.....	9
8.1.2 Transaction Transport Security	12
8.1.3 Public Key Infrastructure.....	12
8.1.4 Public Key Infrastructure encryption methods	13
8.1.5 Secure Socket Layer.....	13
8.1.6 Internet Protocol Security.....	13
8.2 DIRECTORY SERVICES	14
9. DATA INTEGRATION	15
9.1 XML.....	15
9.2 XML SCHEMAS	17
9.3 XML SCHEMA USAGE SCENARIOS	17
10. INFORMATION ACCESS.....	19
10.1 COMPONENT COUPLING AND COHESION.....	19
10.2 MESSAGE QUEUING	20
10.3 OBJECTS, COMPONENTS, SERVICES AND E-SERVICES	20
10.3.1 Objects	21
10.3.2 Components.....	21
10.3.3 Services.....	22
10.3.4 e-Services.....	22
10.4 SIMPLE OBJECT ACCESS PROTOCOL	22
10.5 UNIVERSAL DESCRIPTION, DISCOVERY AND INTEGRATION	23
11. WAY FORWARD.....	24
11.1 GENERAL	24
11.1.1 e-Government XML Namespace	24
11.1.2 e-Government Metadata Framework and Standards.....	24
11.1.3 e-Government Security Framework, Policies and Standards	25

11.1.4	<i>e-Government Message Exchange Patterns</i>	25
11.1.5	<i>e-Government classification for use within UDDI</i>	25
11.1.6	<i>e-Government Presentation Standards</i>	26
11.1.7	<i>e-Government Interoperability Standards</i>	26
11.2	IMPLEMENTATION.....	26
12.	REFERENCES.....	27
13.	MODIFICATION HISTORY	27
14.	MAINTENANCE AND REVIEW CYCLE.....	28
	APPENDIX: ABBREVIATIONS AND ACRONYMS	29

Figures

FIGURE 1:	CONCEPTUAL SECURITY FRAMEWORK FOR E-GOVERNMENT.....	9
FIGURE 2:	INDIVIDUAL ROLE ASSIGNMENT	10
FIGURE 3:	AGENT ROLE ASSIGNMENT	10
FIGURE 4:	ORGANISATIONAL REPRESENTATIVE ROLE ASSIGNMENT.....	11
FIGURE 5:	FOUR-LEVEL AUTHENTICATION MODEL.....	12
FIGURE 6:	DIRECT DATA INTERCHANGE.....	15
FIGURE 7:	TRANSFORMATION OF DATA TO XML	16
FIGURE 8:	DATA INTERCHANGE THROUGH THE USE OF MIDDLEWARE	16
FIGURE 9:	USE OF COMPONENTS IN A TYPICAL 3-TIER ARCHITECTURE	19
FIGURE 10:	DEVELOPMENT MODELS	21

Tables

TABLE 1:	THREE ROLES FOR E-GOVERNMENT USERS.....	9
TABLE 2:	ROLE ASSIGNMENT	10
TABLE 3:	FOUR-LEVEL AUTHENTICATION MODEL	11
TABLE 4:	CONSTRUCTS OF AN XML SCHEMA	17

1. Purpose

The objective of this technical Report is to study Information Communication Technology issues that are specific and particular to the broad area of Electronic Government for Malta (e-Government).

The e-Government Interoperability Framework (IOF) aims to support interconnectivity of heterogeneous and dissimilar Government Information Systems and services. The IOF supports the exchange and use of information between systems and across services. This exchange use of information can be inside and outside Government, and in Malta and abroad. These are high requirements

It has value as a stand-alone document. Additionally, the IOF serves to point to areas where Policies and Standards will need to be developed in support of e-Government. The IOF does not presume, nor should its readers presume, that the IOF is an exhaustive source of all Government ICT Policies, Standards and Directives.

The IOF will provide managers and designers with a reference necessary for the connectivity of similar and dissimilar Architectures within the Government of Malta. The IOF will be used as a product-neutral guide for any new information system or services development, within the ambit of the e-Government Portal. Agents of Government and Third Parties contracted for the development of information systems or services on the MAGNET (MALta Government NETwork) will draw upon the IOF and the Policies, Standards and Directives of CIMU, some of which will derive from it.

2. Who should read this report

This intended audience of this report is all persons contributing in a decision-making role or in a technical role to e-Government in, or on behalf of, the Government of Malta.

- CIMU
- IMOs
- Heads of Department
- Persons procuring ICT within Ministries
- MJLG employees involved in e-Government
- Agents
- Third party services provider

3. Scope of applicability

The scope of applicability of this report is technical and management aspects of ICT interoperability of networks, services and software in relation to e-Government.

4. Definitions

The following terms are defined within the IOF as follows:

Interoperability	Interoperability the ability of two or more systems or components and services to exchange information and to use the information that has been exchanged.
Metadata	Metadata is a definition or description of data. Metadata may, for example describe how, when, and by whom a particular set of data was collected, and how the data is formatted.
Security	Security is the capability of products and of processes to preserve confidentiality, integrity and availability of information and of services. Confidentiality means access allowed to, and only to authorised persons. Integrity means safeguarding the accuracy, completeness of information and processing methods, and availability means access is there when required.
UDDI	Universal Description, Discovery and Integration (UDDI) is an XML-based registry for businesses world-wide to list themselves on the Internet. Its goal is to streamline online transactions by enabling companies to find one another on the Web and make their systems interoperable for e-commerce. UDDI is often compared to a telephone book's white, yellow, and green pages. The project allows businesses to list themselves by name, product, location, or the Web services they offer. Microsoft, IBM, and Ariba spearheaded UDDI.

5. Security classification

Unrestricted

6. Executive summary

The e-Government IOF provides a management and technical and guide for the building of e-Government services as proposed in the Government White Paper on the Vision and Strategy for the Attainment of e-Government.

The IOF tackles the technical aspects of interoperability to facilitate information flow across Government entities and the Public in general. It describes issues that Public Sector organisations need to address for them to restructure their business processes to take advantage of the opportunities provided by increased interoperability.

This IOF is sponsoring Open Standards. These standards serve a wide range of user communities, public and private and transcending national borders, hence the name "open". Open standards will be used to support the interoperability requirements of e-Government. The Government intends to apply such standards to the MAGNET as its ICT backbone for e-Government services provision. Refer to Connectivity to MAGNET (Malta Government Network) Policy (CIMU P 0011:2002), Standard (CIMU S 0011:2002) and Directive (CIMU D 0011:2002).

Open standards are suited to e-Government because all the requirements and possibilities of the latter are not known at the outset.

An operating platform based on Open Standards allows evolution of the solution as more possibilities become known. This emerging solution will be made possible by liberating the underlying platform from particular suppliers and supplier interests which are by definition sectoral and corporate.

As the social, economic, political and technical realities of e-Government become known a platform based on open standards will be developed to accommodate the unfolding requirements. Open standards allow gradual modification and a capability of the architecture to mature in time.

The content of this IOF needs to be revisited and updated on a regular basis to keep the document valid and in line with current ICT practices. The document will also help designers and integrators of systems and of services to achieve interoperability. These systems and services cover software, networks and services. Interoperability will allow Government systems and services to work homogeneously and transparently for the citizens of Malta.

7. Overview

The IOF has been described as an instrument that will support the working together of similar and dissimilar Architectures, that is, heterogeneous information systems and services. Interoperability can be defined as specifications that are relevant to systems interconnectivity, data integration and access to information. Interoperability is a technical and a management issue.

Technical Issues

The study focuses on three technical areas that are essential for interoperability. These are:

- Interconnectivity
- Data Integration
- Information Access.

These areas have been dealt with by considering five major components of the systems that will ultimately provide e-Government services. These five components are relevant to each of the three key areas:

- Applications
- Data Content
- Development Languages
- Services
- Security.

In all of these areas, the main focus of the specification has been to adopt de facto standards, Internet and World Wide Web (W3C, <http://www.w3.org>) standards for all Government information systems and services through which data and services are to be made available over the Internet. There is a need to adopt XML as the main standard for data integration. This strategy should be endorsed by the provision of XML Schemas for use throughout the Government, as is being requested by the strategic partner.

Management Issues

The e-Government IOF must be regarded as a long-term, ongoing initiative that adapts in line with the advancements in ICT practice. Management and design processes need to be put into place to describe how changes to this framework will be managed. These management and design processes regarding the IOF supporting the management and design processes of software, networks and services.

8. Interconnectivity

Interconnectivity is the first step to interoperability. Interconnection is the linkage used to join two or more communications units, such as networks, links, nodes, equipment, circuits, devices, systems and services. [adapted from http://www.its.bldrdoc.gov/fs-1037/dir-019/_2783.htm]. Interconnectivity, with the functionality of interoperability (see definitions section) in the absence of security will in most cases be undesirable or dangerous.

Interconnectivity is based on the following standards:

- Lightweight client software of internet standards (Web Browser)
- IP (Internet Protocol), more precisely IPv4 for the transport service
- HTML (HyperText Markup Language) format accessible in HTTP (HyperText Transfer Protocol)
- Domain Name Server
- Messaging service compatible with the family of SMTP (Simple Mail Transfer Protocol) standards: E/SMTP, MIME
- File Transfer Protocol (FTP)
- Directory compatible with LDAP v3 (Lightweight Directory Access Protocol)
- Transport security SSL v3/TLS.

The external, business requirement is frequently for Interconnectivity to enable the functionality of interoperability, with the added requirement of security.

Security is the capability of products and of processes to preserve confidentiality, integrity and availability of information and of related services. Confidentiality means access allowed to, and only to authorised persons. Integrity means safeguarding the accuracy and completeness of information and processing methods. Availability means access is there when required. These related concepts clarify the preceding definition of security.

8.1 Security

Security is critical for the success of e-Government as it enables the trust and confidence in its services. The security e-Government means that one must ensure:

- Data Confidentiality
- Data Integrity
- Citizen identification
- Non-repudiation.

Thus e-Government raises transaction security concerns and thus the success of e-Government depends on the capability to guarantee that the transaction environment is available, reliable and secure.

Security implementation will observe all the requirements defined in the Electronic Commerce Act. Furthermore, measures need to be taken so that all e-Government applications will be compliant with the Data Protection Act.

The security framework being proposed for the e-Government is divided into two areas:

- Citizen identification, authentication and authorisation. This is the method of ensuring citizen identification and non-repudiation.

- Transaction Transport Security. This incorporates methods for securing data during transition to ensure data confidentiality and integrity.

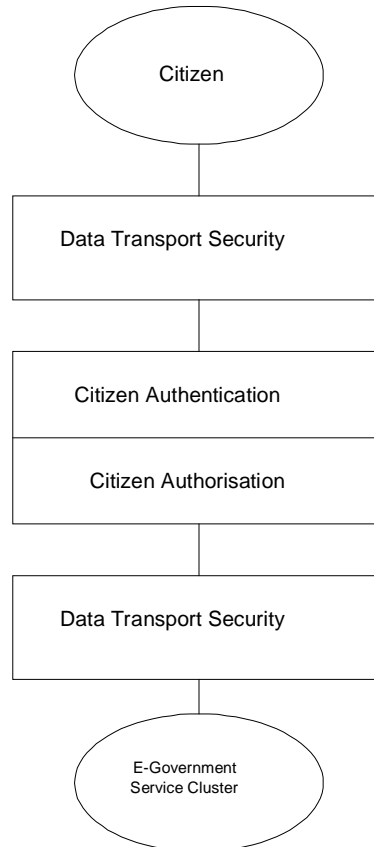


Figure 1: Conceptual Security Framework for e-Government

8.1.1 Citizen Identification, Authentication and Authorisation

Citizen authentication is the process of verifying and confirming the identity of the person accessing data or services. Authorisation is the process of allowing access to the data or services, conditioned by the individual’s access levels. Within the e-Government framework, the users who will be using the services offered via the portal can assume one or a combination of the following three roles:

Individual	An individual can access data or services on his own behalf.
Agent	An agent is an individual or organisation that can access data or services on behalf of another individual/s or organisation/s provided that consent is granted to the agent by the data subject (an individual or organisation).
Organisational Representative	An organisational representative is an individual who can access data or services on behalf of an organisation/s provided that consent is granted to the data subject (representative by the organisation).

Table 1: Three Roles for e-Government Users

		Assumes the Role of	When acting on behalf of	Example
1	Individual	Individual	Himself/herself	
2	Individual	Agent	Another Individual	Legal procurator
3	Organisation	Agent	Individual	Insurance agency
4	Organisation	Agent	Another Organisation	Auditing firm, Company doctors
5	Individual	Organisational Representative	Organisation	Employee, Accountant

Table 2: Role Assignment

The roles identified and described above are an initial logical grouping of users and are subject to change. This model works best in a B2C environment (in this case, G2C) and will need modification when considering G2G and G2B scenarios.

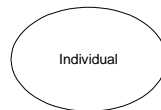


Figure 2: Individual Role Assignment

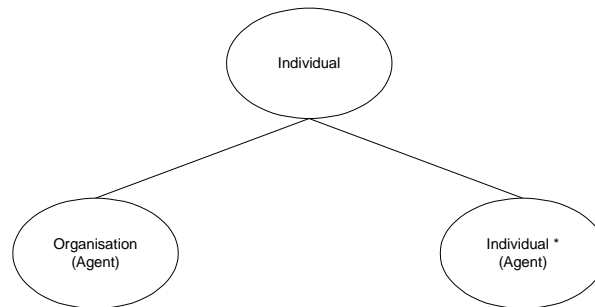


Figure 3: Agent Role Assignment

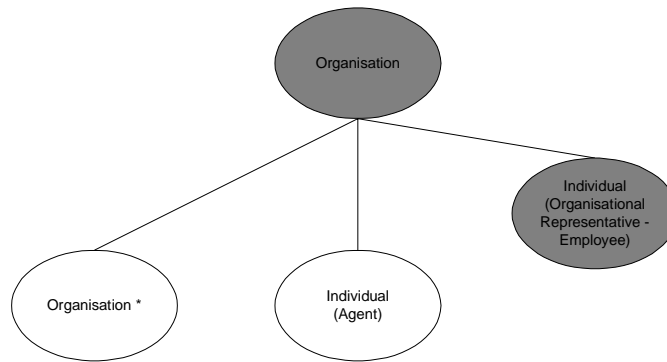


Figure 4: Organisational Representative Role Assignment

The data and services to which an individual has access at any time whilst assuming any of the roles listed above will be determined by his or her profile which will maintained by the Directory Services.

In view of this, and of the nature of the data and services being offered within e-Government, a four-level authentication model can be established. Different authentication levels will allow different user roles to access different services. The four levels are in increasing order of security, denoted as L0, L1, L2 and L3. Below is a table, with examples, depicting this four-level authentication model.

Name of authentication	Level	Explanation	Example
No authentication	L0	Data or services intended for the general public. No authentication will be required.	Viewing of Maltese Laws
Restricted authentication	L1	Documents or services of certain importance. Authentication will be required to protect against misuse or loss.	Viewing of status of a particular court case
Confidential authentication	L2	Personal documents or services affecting personal data. Authentication will be required to protect against significant problems or losses.	
Maximum authentication	L3	Strictly confidential personal data or financial transactions. Authentication will be required to prevent considerable financial loss or serious political damage to Government.	Viewing individual's medical records

Table 3: Four-level Authentication Model

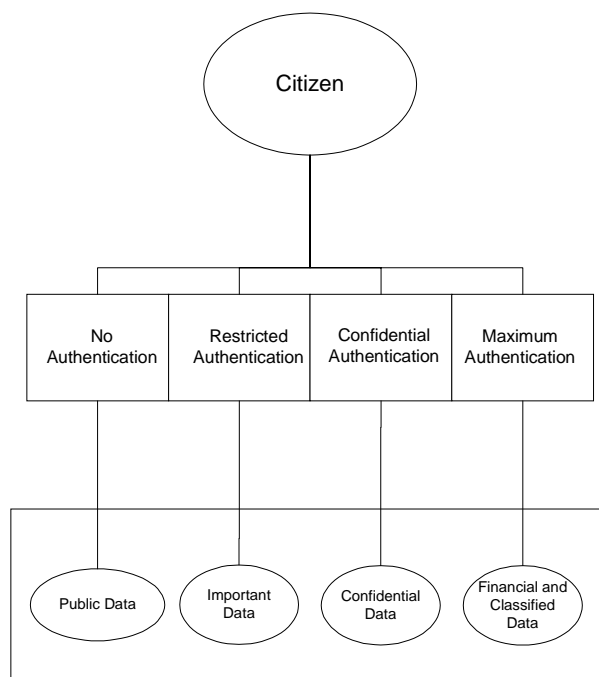


Figure 5: Four-Level Authentication Model

8.1.2 Transaction Transport Security

As data or personal information is passed from one entity to another via a medium, there is always the risk that this data is intercepted. Consequently it may be stolen, misused, modified or denied (non-repudiation). This means jeopardising the confidentiality and/or integrity of the data as well as undermining trust in e-Government.

To prevent such methods of security attack, it is proposed that a Public Key Infrastructure (PKI) will be used within the e-Government framework. PKI will enable us to:

- Correctly authenticate users.
- Effectively encrypt the data before it is passed over a public network with the minimal risk of the same data being intercepted and interpreted.

8.1.3 Public Key Infrastructure

A Public Key Infrastructure (PKI) enables users of a basically unsecured public network such as the Internet to securely and privately exchange data and money through the use of a public and a private cryptographic key pair that is obtained and shared through a trusted authority. The public key infrastructure provides for a digital certificate that can identify an individual or an organisation and the directory services that can store and, when necessary revoke certificates.

The public key infrastructure assumes the use of **public key cryptography**, which is the most common method on the Internet for authenticating a message sender or encrypting a message.

Traditional cryptography has usually involved the creation and sharing of a secret key for the encryption and decryption of messages. This secret or private key system has the significant flaw that if the key is discovered or intercepted by someone else, messages can be easily decrypted. For this reason, public key cryptography and the public key infrastructure is the preferred approach on the Internet.

A public key infrastructure consists of:

- A certificate authority (CA) that issues and verifies a digital certificate. A certificate includes the public key or information about the public key
- A registration authority (RA) that acts as the verifier for the certificate authority before a digital certificate is issued to a requestor
- One or more directories where the certificates (with their public keys) are held
- A certificate management system
- An encryption method.

8.1.4 Public Key Infrastructure encryption methods

There are various methods to implement a Public Key Infrastructure, PKI environment. Although the methods are different in nature (i.e. some are application based and others are network based) they use the same digital certificate mechanism to encrypt the data before it is transmitted over a public network. Two of the most common methods used are Secure Socket Layer (SSL) and Internet Protocol Security (IPsec).

8.1.5 Secure Socket Layer

Secure Socket Layer (SSL) is a commonly used protocol for managing the security of a message transmission on the Internet. It is an application based transport security mechanism. SSL uses a program layer located between the Internet's Hypertext Transfer Protocol (HTTP) and Transport Control Protocol (TCP) layers. SSL is included as part of most Internet web browser products. SSL uses the public-and-private key encryption system, which also includes the use of a digital certificate. If a Web site is hosted on a server that supports SSL, SSL can be enabled and specific Web pages can be identified as requiring SSL access.

8.1.6 Internet Protocol Security

Internet Protocol Security, IPsec is an evolving standard for security at the network or packet processing layer of network communication. Earlier security approaches have inserted security at the application layer of the communications model. IPsec will be especially useful for implementing virtual private network and for remote user access through dial-up connection to private networks. A big advantage of IPsec is that security arrangements can be handled without requiring changes to individual user computers.

8.2 Directory Services

The above framework for the identification, authentication and authorisation of the citizen can be achieved by implementing Directory Services. These are services provided to individual authenticated users to enable them to make use of various services offered by the operating environment such as printing, e-mail and file access. The Directory services also enable the management of users listed in that directory.

X.500 is fundamental standard for Directory Service. X.500 is a way to develop electronic directories that it can be part of a global directory available to anyone on the Internet. [adapted from http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213397,00.html]

Many organisations have implemented an X.500 directory. Because these directories are organised as part of a global directory, one can search for thousands of entries from the World Wide Web. The X.500 directory is organised under a common "root" directory in a "tree" hierarchy.

An entry at each of these levels must have certain attributes; some can have optional ones established locally. Each organisation can implement a directory in its own way as long as it adheres to the basic schema or plan. The distributed global directory works through a registration process and one or more central places that manage many directories. Providing an X.500 directory allows an organisation to make itself and selected content known on the Internet. There are less full implementations of X.500 termed LDAP, (Lightweight Directory Access Protocol).

LDAP is a software protocol to enable anyone to locate resources such as files and devices in a network, whether on the public Internet or on a corporate intranet. LDAP is a "lightweight" (smaller amount of code) version of Directory Access Protocol (DAP), which is part of X.500. Netscape, Microsoft and CISCO support LDAP.

In a network, a directory gives information on where in the network something is located. On TCP/IP networks the Domain Name System (DNS) is the directory system used to relate the domain name to a specific network address.

The domain name may in some cases not be known. LDAP allows search without knowing where in the directory the entity is located. An LDAP directory is organised in a simple "tree" hierarchy consisting of the following levels:

- The root directory, which branches out to
- countries, each of which branches out to
- organisations, which branch out to
- organisational units (divisions, departments, and so forth), which branches out to (includes an entry for)
- Individuals.

Individuals include people, files, and shared resources such as printers.

An LDAP directory can be distributed among many servers. Each server can have a replicated version of the total directory that is synchronized periodically.

9. Data Integration

A data integration effort refers to the use of data from a number of different sources. Data integration refers to data and documents. Much of the power of data and document integration techniques lies in their ability to associate like data from a variety of different sources. XML is a language to describe data.

9.1 XML

XML products will be written so as to comply with the recommendations of the World Wide Web Consortium (W3C, <http://www.w3.org>). Although XML provides a common format for data interchange, its use must only be resorted to when needed. In those cases where the data component interfaces are known, one need not incur the overhead computational cost of transforming data to XML and back as opposed to exchanging data directly.

The use of XML-enabled middleware will simplify the development of data access components, especially when the number of data sources involved in data interchange increases. However, if the use of such middleware is deemed appropriate, systems must be designed with the use of these products in mind. The use of middleware conditions how data access components are designed and if a middleware product is introduced after a system has been designed and developed, the data access components will need to be changed.

In view of this, below are three possible data interchange scenarios:

1. **Direct interchange**, where data is passed in native format.

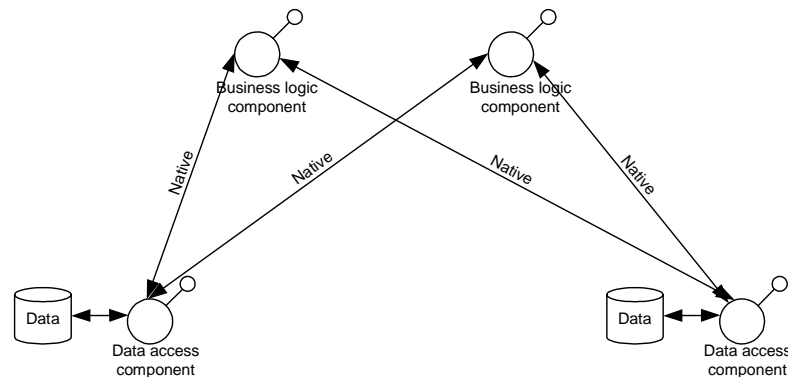


Figure 6: Direct Data Interchange

2. **Transformation of data to XML** in the case of data being passed between disparate systems.

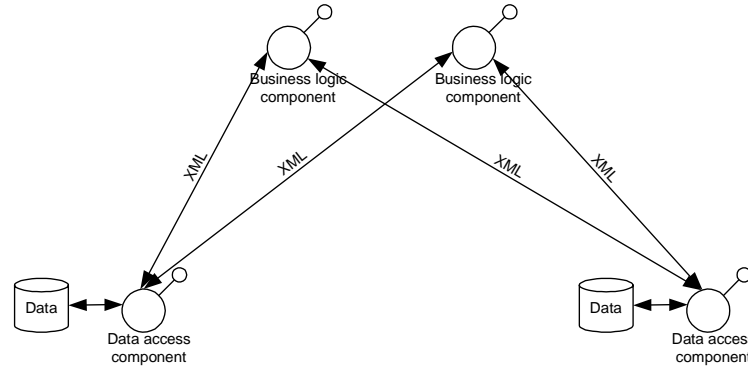


Figure 7: Transformation of Data to XML

3. **Through the use of middleware** to handle the data interchange, especially where a large number of data sources are involved which potentially reside in different geographical locations.

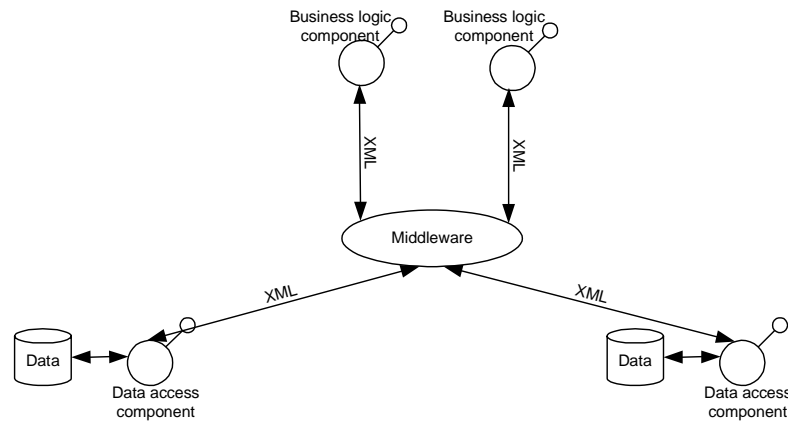


Figure 8: Data Interchange through the use of Middleware

9.2 XML Schemas

The purpose of an XML schema is to define and describe a class of XML documents by using the following constructs to constrain and document the meaning, usage and relationships of their constituent parts:

Data types	Provide for primitive data typing, including byte, date, integer, sequence, etc.
Entities	An XML document may consist of one or many storage units, called entities. They all have content and are all identified by name.
Elements	An element can contain text, other elements, a mixture of text and elements, or nothing at all.
Attributes	Attributes are used to assign values to elements, including default values.
Notations	Notations identify by name the format of certain entities and elements, or the application to which a processing instruction is addressed.

Table 4: Constructs of an XML Schema

Schema constructs may also provide for the specification of implicit information such as default values. Schemas document their own meaning, usage, and function. Thus, the XML schema language can be used to define, describe and catalogue XML vocabularies for classes of XML documents.

9.3 XML Schema Usage Scenarios

The following usage scenarios describe XML applications that should benefit from XML schemas. They represent a wide range of activities and needs that are representative of the problem space to be addressed. They are intended for use during the development of XML schemas as design cases that should be reviewed when critical decisions are made.

1. **Publishing:** Distribution of information through publishing services. This involves collections of XML documents with complex relations among them. Structural schemas describe the properties of headlines, news stories, thumbnail images, cross-references, etc.
2. **Electronic commerce transaction processing:** Libraries of schemas define business transactions within markets and between parties. A schema-aware processor is used to validate a business document, and to provide access to its information set.
3. **Supervisory control and data acquisition:** The management and use of network devices involves the exchange of data and control messages. Schemas can be used by a server to ensure the validity of outgoing messages, or by the client to allow it to determine what part of a message it understands. In a multi-vendor environment, the server can discriminate between data governed by different schemas (industry-standard, vendor-specific). The server will know when it is safe to ignore information it does not understand and when, on the other hand, it should raise an error. Applications include security systems and process control.
4. **Traditional document authoring and editing governed by schema constraints:** One important class of application uses a schema definition to guide an author in the development of documents. A simple example might be a memo, whereas a more sophisticated example is a complex request form. The application can ensure that the author always knows what to enter, and might even ensure that data entered is valid.
5. **Query formulation and optimisation:** A query interface inspects XML schemas to guide a user in the formulation of queries. Any given database can emit a schema of itself to inform other systems what can be considered as legitimate and useful queries.

6. **Open and uniform transfer of data between applications and databases:** XML has become a widely used format for encoding data (including metadata and control data) for exchange between loosely coupled applications. The representation of the data exchange by XML Schema definitions simplifies the task of mapping the data exchange to and from application internal data models.

7. **Metadata Interchange:** There is growing interest in the interchange of metadata (especially for databases) and in the use of metadata registries to facilitate interoperability of database design as well as DBMS, query, user interface, data warehousing, and report generation tools.

A point needs to be made about metadata. Metadata is a definition or description of data. Both a data item and data about it are data, the latter being metadata. Metadata may, for example describe how, when, and by whom a particular set of data was collected, and how the data is formatted. Metadata is essential for understanding information and its structure. An information system for a library can contain information (metadata) about publications (data). A file system maintains permissions (metadata) about files (data).

Software functions used in different Government applications having common data will be encapsulated into business logic components that are common to several applications. For these components to be used to their maximum benefit, this exercise should be complemented by the production of XML based data schemas relating to these common business functions. These can be reused across Government applications to reduce the costs and risks of developing data interchange systems. It is these common business functions which should be considered first for the production of XML schemas.

10. Information Access

10.1 Component Coupling and Cohesion

Coupling refers to the way data is exchanged between components. Loose coupling is generally better than tight coupling. The loosest, and therefore preferred, type of coupling is data coupling, where data is transferred as parameters via well-defined interfaces. The tightest, or least desirable, coupling involves components directly referencing shared variables. Tight coupling often indicates that components are not insulated from each other, and are not designed to be separate and independent. Tightly coupled components are usually complex, difficult to maintain, and monolithic. As a result there is very little flexibility regarding physical distribution of components. Two applications that communicate with each other via message queues, but which are otherwise independent of each other, would be considered loosely coupled.

Cohesion reflects the degree to which one component implements one function or a group of similar functions. For example, cohesive components do not implement multiple, disparate services, such as presentation and application logic. Highly cohesive components are typically more understandable and thus easier to maintain. Additionally, cohesion promotes logical and physical software distribution flexibility, which in turn promotes system scalability. An application composed of logically separate presentation, application, and data management components, would be considered highly cohesive.

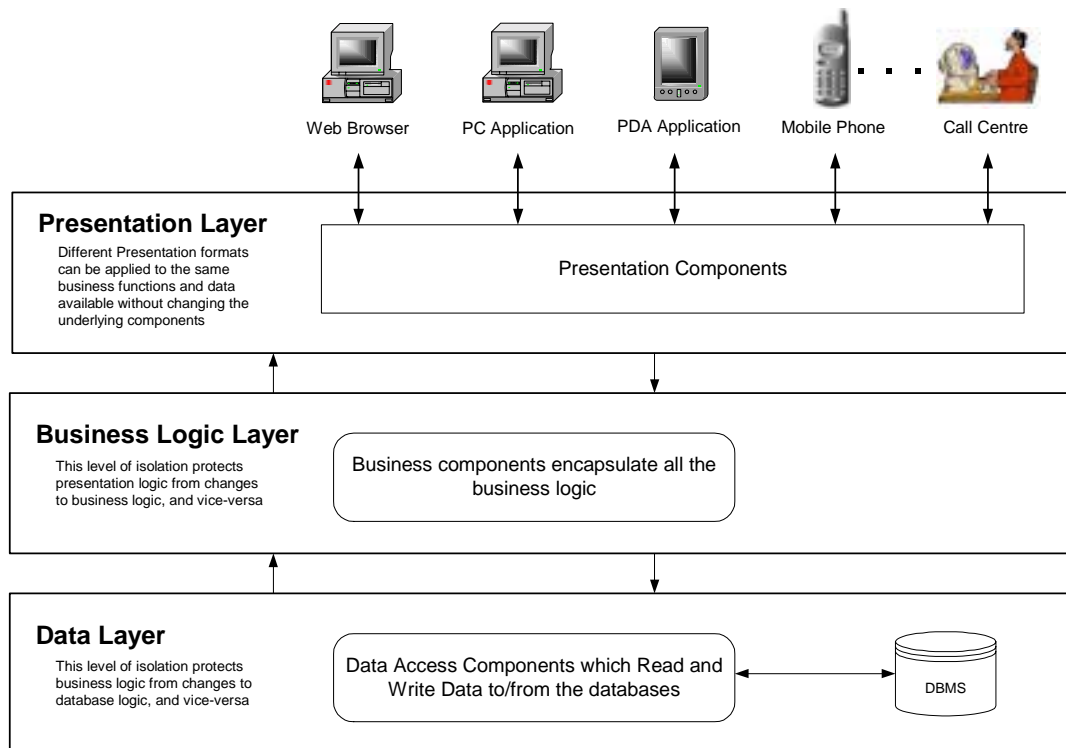


Figure 9: USE OF COMPONENTS IN A TYPICAL 3-TIER ARCHITECTURE

Software coupling and cohesion directly affect software modularity and interface design. As a result, coupling and cohesion directly affect the flexibility and complexity of software architectures. For example, when software component interfaces are based on widely accepted standards, as illustrated below, logical software tiers can promote component interoperability and substitutability. That is, logical tiers can allow components within one tier to be changed without affecting other tiers.

10.2 Message Queuing

Message queuing enables distributed applications to reliably exchange mission-critical data regardless of hardware, operating system, or available connectivity.

Message queuing is roughly analogous to an e-mail system. When an e-mail message is sent, the note is addressed and sent to the intended recipients. One is not usually concerned about the underlying delivery route of the e-mail message or when the recipients pick it up. Likewise, one can log into an e-mail server and pick up messages at his or her discretion without maintaining a direct link with those who have sent the e-mail.

Message queuing works in much the same manner as an e-mail system, except that applications (not people) are sending data (not notes). Like e-mail, the sending application does not have to be concerned about delivery routes or when the receiving application will pick up the message. The receiving application can pick up new messages whenever it is appropriate without necessarily maintaining a direct link with the sending application.

10.3 Objects, Components, Services and e-Services

The scene of application development has seen a transition of its design and development paradigms, from objects to components and more recently services and e-services. However, each remains a valid development solution, having benefits to give in different circumstances.

The design and development of systems should have as their main focus the provision of services. Service Oriented Architecture (SOA) has emerged as the best practice for systematic logical design of applications, offering greater reuse and more access to the business functions, or logical services, of the application from other applications. SOA is a logical architecture where definitive business functions of the application are exposed for programmatic access via a well-defined formal interface, with some means of identifying and locating the function and the interface when it is needed. SOA has been implemented both via a tightly coupled request/reply model and via a loosely coupled messaging model. It has also been implemented using either an object request broker (ORB) or messaging middleware. However, SOA services are typically intended for external (heterogeneous) access. Thus, the messaging model or messaging middleware are typically preferred for their implementation.

Each of these four programming styles may use either a tight or a loose coupling model. However, the farther out along the x-axis, the link between programs typically becomes looser and the use of the loosely coupled messaging model and the messaging middleware become more beneficial. Conversely, the closer to the root of the diagram, the more beneficial is the use of a tightly coupled request/reply programming model and of the ORB-style middleware.

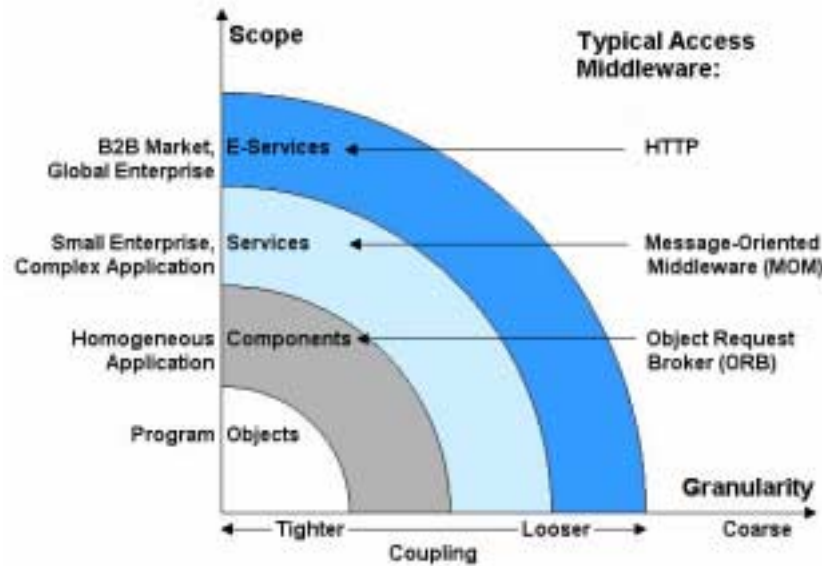


Figure 10: Development Models

(Source: Gartner)

10.3.1 Objects

Objects are a natural evolution of the best programming practice: modularity. Long before object-oriented (OO) programming became mainstream, software engineers re-used software. Prior to OO, software engineers developed subroutines, I/O modules and include files to encapsulate fragments of application logic for reuse. This re-use was within one application and across several applications.

The OO style of modularity is now mainstream practice. Most new applications are developed, at least in part, with the use of an OO programming platform (e.g. Java, and C++). However, an object's scope of visibility is limited to the internal resources of a program, where it is instantiated. Object methods are invoked only from within the program. Objects are incorporated in the executable program so that the calling program and the object are more than tightly coupled; they are one.

10.3.2 Components

Scalable applications must be able to run across different servers. Thus, some of the methods may have to be invoked remotely. The caller and the server can not be one in this case. To support remote invocation of object methods, the industry had invented ORBs. These programs, accessed via an ORB, are components. The purpose of an ORB is to allow remote access to an object method with as little intrusion on the program or the programming model as possible. Thus, components are typically used in a tightly coupled, request/reply environment. This applies to all currently relevant component models, i.e., Distributed Component Object Model (DCOM), Common Object Request Broker Architecture (CORBA) and Java Remote Method Invocation (RMI).

The scope of components is bigger than that of objects, because they are visible to the remote programs. Many object methods are typically invoked on behalf of a single, remote procedure call (RPC). Components do not replace objects, but rather are built on top of them and component specifications are likely to support both the tightly coupled and the loosely coupled programming models.

It is a reality in application development that parts of applications will need to be rewritten because of changing requirements, legislation, changes in Government policies, etc. It is

therefore very important to design components with business (application) logic being kept separate from data access. Separation of business logic from the presentation layer, as in client server architectures, is an added benefit.

10.3.3 Services

Services represent definitive published business functions of an application. They can be implemented using a tightly coupled request/reply programming model or a loosely coupled messaging programming model. Tightly coupled services operate just like components, but are published to a target audience potentially consisting of heterogeneous client platforms.

The loosely coupled services are designed differently, using the messaging model. Messaging is the preferred model for services, given that they are typically invoked from other applications. The greater the degree of heterogeneity, the more likely a loosely coupled model would work better. Loosely coupled services are designed to operate independently of their callers. The corresponding applications may run on different machines, on different application platforms and in different geographical regions.

10.3.4 e-Services

Business-to-business (B2B) interactions have become very important in today's information economies, in which both the business and technology differences are greater, the distances are longer and the possibilities for any co-ordination are further reduced.

Here, the program topology is likely to be loosely coupled in most cases. Enterprises are also not likely to have available the same proprietary messaging middleware. Thus, services offered across enterprises will tend to rely on existing messaging middleware and a standard method for formatting messages. Typically, this will be Extensible Markup Language (XML) messaging over HTTP or SMTP transports.

e-services are services deployed over a universal Internet transport in an Internet-standard format, such as XML over HTTP. Technically, any service may be converted to an e-service by routing it over HTTP and arranging the messages into an XML schema. Logically, however, e-services represent a separate domain of functional content, intended, authorised and advertised specifically for B2B access.

10.4 Simple Object Access Protocol

Simple Object Access Protocol, SOAP allows the exchange of information in a decentralised, distributed environment using XML, making it usable in a large variety of systems ranging from messaging systems to Remote Procedure Calls (RPC). SOAP facilitates interoperability among a wide range of programs and platforms, making applications accessible to a broader range of users. It also combines the proven Web technology of HTTP with the flexibility and extensibility of XML. Existing applications would need to be modified to accommodate this.

10.5 Universal Description, Discovery and Integration

Universal Description, Discovery and Integration (UDDI) is an XML-based registry for businesses world-wide to list themselves on the Internet. Its goal is to streamline online transactions by enabling companies to find one another on the Web and make their systems interoperable for e-commerce. UDDI is often compared to a telephone book's white, yellow, and green pages. The project allows businesses to list themselves by name, product, location, or the Web services they offer. Microsoft, IBM and Ariba spearheaded UDDI.

A "Web service" is specific business functionality exposed by a company through an Internet connection, for the purpose of providing a way for a third party to use the service. The Universal Description, Discovery and Integration (UDDI) specifications define a way to publish and discover information about Web services. UDDI takes an approach that relies upon a distributed registry of businesses and their service descriptions implemented in a common XML format. Programs and programmers use the UDDI Business Registry to locate information about services and, in the case of programmers, to prepare systems that are compatible with advertised Web services or to describe their own Web services for others to call.

11. Way Forward

11.1 General

The IOF is a document that requires regular update to keep it relevant. It is therefore being proposed that this document must be revised after 6 months and subsequently yearly. Certain Policies and Standards arise from IOF requirements and these constitute next steps for CIMU.

Against the background depicted above, CIMU, and the Architecture Unit in particular, will need to work closely and on an ongoing basis with the e-Government Joint Venture (JV). Together CIMU and the JV need to co-develop the following Policies and Standards. This joint standards development is expected to take ICT through 2002 and 2003, for the benefit of the evolution of e-Government over the coming seven years. This Policies and Standards co-development is expected to start as soon as the JV contract is in place. The following text is an extract taken from the Maltese e-Government Framework Architecture version 0.4.

(extract from the Maltese e-Government Framework Architecture Overview)

The local Maltese Government standards are required to support the e-Government. Each department and service connecting into the e-Government framework must comply with these local standards to ensure data interoperability, ease of integration and maintaining security.

11.1.1 e-Government XML Namespace

To ensure interoperability of data a centrally agreed XML Namespace is required. This XML namespace contains the data model of the commonly used data elements, such as date, citizen's name, and address. For each object within the data model the encoding standards are produced which include the list of attributes, lists of acceptable values and patterns to be matched. Once the data model has been completed the task of creating the individual business messages to be communicated between components is simplified. To facilitate interoperability with businesses and other foreign government organisations international data standards will be used where available.

The process of generating the e-Government XML Namespace is best achieved by the use of Working Groups made up of personnel from a number of different departments.

11.1.2 e-Government Metadata Framework and Standards

To facilitate the creation of knowledge management across Government a single metadata framework is required. This metadata framework ensures that knowledge is categorised in a consistent and searchable format for all types of information in all departments.

To ensure interoperability with other organisations and comply with international standards the following two standards should be considered.

Dublin Core (Government) Working Group

The Dublin Core is a set of metadata elements that has been agreed following international discussion with businesses and governments. The Dublin Core should be implemented in its entirety, although the Maltese Government can add additional elements as well as restrict or extend the allowable values.

Dublin Core with e-Record Management Extensions

As the Maltese Government has both current information / knowledge as well as existing records the Dublin Core should also be extended to include electronic Records Management tags, such as location of electronic record, data to de-classification.

11.1.3 e-Government Security Framework, Policies and Standards

To ensure that the e-Government Framework is secure and also perceived to be secure an e-Government Security Framework is required. This framework will be based upon existing international standards and best practice, and will ensure that there is a single end-to-end security model for the e-Government Framework. The Security framework should cover areas such as data security, server security, personnel vetting and physical security.

The Security Framework should be developed by a cross-Government team of Security Officers with outside assistance as required.

The e-Government Security Framework will comply with Malta's ISO 17799:2000 standards.

e-Government Authentication Framework and Standards

The e-Government Authentication Framework and Standards will be a sub set of the Security Framework and will specify the authentication credentials to be used by citizens and government users.

e-Government Authorisation Framework and Standards

The e-Government Authorisation Framework and Standards will be a sub set of the Security Framework and will specify the Authorisation standards and Authorisation tokens to be used by the Services exposed within the e-Government Framework. The Authorisation framework will include the levels of Authentication credential required for Authorisation levels.

11.1.4 e-Government Message Exchange Patterns

To ensure that the Messages are transported in a reliable manner and in the correct order the e-Government Message Exchange Patterns Standards will be required. The Standards would also include routing elements and message id elements to be inserted into the SOAP Envelope header.

There are currently no internationally recognised Web Service Standards, although there are a number of initiatives currently in progress to define these.

11.1.5 e-Government classification for use within UDDI

To ensure that UDDI directory can be searched in a consistent manner and that all services are described in a consistent manner, an e-Government classification (taxonomy) is required as an extension to the base classification (taxonomy) within the UDDI. The e-Government classification (taxonomy) will include Malta explicit location data and Government Functionality Terms.

e-Government UDDI Service Operation Standards

The e-Government UDDI Service Operation Standards will be defined to ensure that all services exposed into the e-Government Framework are available and the definition of the interfaces are not redefined unnecessarily. This will ensure that the business processes based upon the services are stable.

e-Government Message Tracking / Query Framework and Standards

The e-Government Message Tracking / Query framework and standards are required to ensure the progress and status of a service request or response can be reported irrespectively of where in the framework the message currently resides. The audit record is sufficiently detailed enough to ensure that the legal requirements of proof of delivery are met.

11.1.6 e-Government Presentation Standards

To ensure the creation of perception of joined-up government all the Portal rendering complies with the e-Government Presentation Standards.

11.1.7 e-Government Interoperability Standards

The e-Government Interoperability standards are defined to ensure all electronic communication formats between the government and citizen, as well as between government departments can be interpreted by the receiving entity. This includes details such as the format for electronic textural document transfer.

11.2 Implementation

The implementation of these ideas is a longer term, multiyear effort.

Organisationally the tasking of CIMU with defining and maintaining the Architecture means that the responsibility is formally located within a government entity.

The next step, as discussed above, is to strengthen the CIMU organisational chart so that any risks of over-dependence on outsourcing by CIMU is avoided. This has been discussed in the CIMU Business Plan for 2002.

As discussed above, interaction between the JV and CIMU will need clarification and development of the relationship.

With these issues being attended to, and recalling the choice of Open Standards, Government will need to proceed to implement on the basis of the following principles:

1. the avoidance of vendor-specific choices
2. the favouring of solutions based on technologies that are in the process of being standardised
3. the favouring of solutions that are open, modular and evolutionary

Measures will need to be taken by the arm of CIMU tasked with compliance to see that actual projects are in fact being run according to the Architecture and the Standards embraced in this document.

12. References

- A New Zealand E-Government Interoperability Framework (e-GIF), Version 0.9 Release 8, Feb 2002, New Zealand Government State Services Commission
<http://www.e-government.gov.nz>
- Architecture Guidelines for Trans-European Telematics Networks for Administrations Part 1 – Generic Guidance Version 5.3, author Enterprise DG/B/5, February 2001
<http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showDocument&parent=highlights&documentID=295>
- e-Government interoperability framework, Version 3, Office of the e-Envoy, UK
<http://www.govtalk.gov.uk/interoperability/gif.asp>
- Extensible markup language (XML) 1.0 (Second Edition), W3C Recommendation 6 October 2000
<http://www.w3.org/TR/REC-xml>
- Gartner
<http://www3.gartner.com/lnit>
- Generic Interoperability Framework, Sergey Mernik et. Al. Department of Computer Science, Stanford University
<http://www.digilib.stanford.edu/digilib/ginf/WD/ginf-overview/>
- Hypertext Transfer Protocol – HTTP/1.1, Jan 1997
<http://www.cis.ohio-state.edu/cgi-bin/rfc/rfc2068.html>
- Schools Interoperability Framework Implementation Specification, Version 1.0, Revision 1 Final Version August 31, 2001
<http://www.sifinfo.org/spec.html>
- XHTML 1.0: The Extensible Hypertext Markup Language, A Reformulation of HTML 4 in XML 1.0, W3C Recommendation 26 January 2000
<http://www.w3.org/TR/xhtml1/>
- XML Schema Part 1: Structures, W3C Recommendation 2 May 2001
<http://www.w3.org/TR/xmlschema-1/>

13. Modification history

Version	Date	Changes
1.0	11.06.2002	Initial Release

14. Maintenance and review cycle

The maintenance and review cycle of this Report is set for six (6) months after the initial release as indicated in the effective date. Subsequent maintenance to this Report will be based on a twelve (12) month cycle.

Appendix: Abbreviations and acronyms

B2B	Business to Business (e-commerce)
B2C	Business to Customer (e-commerce)
CIMU	Central Information Management Unit
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
D2D	Department to Department (e-Government)
G2C	Government to Citizen (e-Government)
G2G	Government to Government (e-Government)
HTTP	HyperText Transfer Protocol
ICT	Information and Communication Technology
IOF	Interoperability Framework
IPSec	Internet Protocol Security
ITF	IT Facilitator
PKI	Public Key Infrastructure
RMI	Remote Method Invocation
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TCP/IP	Transport Control Protocol / Information Protocol
UDDI	Universal Description Discovery and Integration
XML	eXtensible Markup Language