International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 4, Issue 1, January 2015 ISSN 2319 - 4847

Requirements Engineering of Interoperability Middleware Design for E-Healthcare Services in Developing Countries

Lala, O.G [1], Emuoyibofarhe J.O [2] and Oladosu J.B [3]

[1] Department of Computer Science and Information Technology,
Bowen University, PMB 284, Iwo, Osun State, Nigeria

[2] Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, PMB 4000, Ogbomoso, Oyo State, Nigeria

[3] Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, PMB 4000, Ogbomoso, Oyo State, Nigeria

ABSTRACT

The deployment environment requirements of e-healthcare operation in Nigeria were puts into perspective in the interoperability middleware design in order for the middleware to work successfully where it is deployed. This includes the requirements shaped by the infrastructure available at the deployment area which are; network, electric power supply, finances, device availability etc. It further explores issues such as users' and structural readiness to adopt new technologies made available to them for use and the culture of the deployment environment. The interoperability middleware design presents the resource case that highlights the resource constraints that have to be put into consideration during the middleware's development. The resource case enables operational decisions that will make the system function with the available resources when deployed. Existing middleware technologies were selected and comparative evaluation was done on them using quality requirements. The design of the middleware operation was done using: Simple Object Access Protocol (SOAP), Web Service Definition Language (WSDL) and Universal Description, Discovery and Integration (UDDI). The interoperability operation was designed using: Extensible Markup Language (XML), Entity, XML Schema Library, Service Provider, Service Requestor and Service Registry. The designed middleware took into consideration two functional applications of the middleware for interoperability, though multiple applications can exist on the middleware. These applications are: i. Electronic Health Record (EHR) and ii. Medical Billing System (MBS). Mobile Broadband in form of GPRS, 3G communication is recommended as the internet connectivity option for implementation. This is due to its minimal cost and effort to set up. This is minimal due to the existing massive investment in telecom infrastructure by telecommunication companies in Nigeria.

Keywords:- Middleware, Interoperability, Service Provider, Service Requestor, Service Registry, Requirements

1. INTRODUCTION

In order for healthcare industry to maintain and improve quality of healthcare and to achieve e-healthcare integration, it has to depend on Information Technology (IT) [1]. This is because ICT in healthcare initiatives increases the ability to meet challenges in the provision of healthcare services. Subsequently, it would reduce costs to the system by eliminating repeated diagnostic testing, redundant record keeping, and high travel costs associated with seeking and receiving healthcare in centers located far from patients. It could therefore reduce or prevent adverse patients' outcome [2]. Information and Communication Technology (ICT) can undoubtedly improve the use of information by healthcare professionals in day-to-day patient care [3]. The overwhelming acceptance of ICT has culminated into a number of fields such as e-Healthcare [4]. However, ICT projects in many developing countries are generally associated with failure, mainly due to operational, contextual and strategy constraints. E-Healthcare represents a substantial ICT investment and as such, its failure can result in huge losses in time, money and effort [5]. The role of ICT can no longer be ignored within the healthcare industry. In fact, for the healthcare industry to maintain and improve both clinical and business operations, it has to depend on Information Technology (IT) [6]. This is as a result of the capability of e-health initiatives ability to tackle challenges that exist within the healthcare industry [2]. Electronic health (e-health) describes the application of ICT across a whole range of functions that affect the healthcare industry when it comes to matters relating to health through the various solutions that exist [7]. E-health can also be described as any electronic exchange of health related data through an electronic connectivity for improving efficiency and effectiveness of healthcare delivery. The growth of the internet and deployment of personal computers has really facilitated the growth in telemedicine use [8]. Telemedicine can provide learning opportunities to the doctors and nurses in the rural areas and also provide a platform for second opinions among professionals. Additionally, it can save patients and physician's time and money as they will not have to travel far distances to provide or receive hospital services. Moreover, it can allow underprivileged rural hospitals to share equipment and human resources within well-equipped hospitals [9], [10]. ICT projects are associated with failure. E-healthcare represents a substantial ICT investment, and as such, failure of e-Healthcare systems can result in huge losses in time, money, and effort [11]. It is necessary that all e-Healthcare stakeholders have the tools and mechanisms to understand the readiness concept, and to determine the readiness status of communities before implementing costly e-Healthcare innovations [12]. Requirements engineering is concerned with the goals, desired properties and constraints of complex systems that involve software systems, organizations and people. It also covers how requirements relate to business processes, soft issues, work redesign, system and software architecture and testing. This process is regarded as one of the most important aspects of building an information system because it is during this process that it is decided what is to be built, [13], [14]. Middleware products enable information to be shared in a seamless real-time fashion across multiple functional departments, geographies and applications. In literature, different approaches to middleware selection can be found, [15]

2. Resource Case

The resource case highlights the resource constraints that have to be put into consideration during the middleware's development. The resource case enables operational decisions that will make the system function to be made with the available resources when deployed. This is shown in Figure 1. The need to meet the resource constraints highlighted in the resource case led to the consideration of a number of operational models. It could however be concluded that to meet the work under the available resources, the developed middleware should run as a service, commonly tagged Platform as a Service (PaaS). The deployed system runs in a Service Oriented Architecture (SOA), thus providing interoperability and information exchange to a broad range of applications in a highly scalable manner. This platform as a service based middleware can be provided by government agency, private / public organization. E-health interoperability services are then provided from the cloud as a service. This aids the deployment of the middleware in a centralized location and applications and devices utilize it on an On-Demand basis rather than the individual middleware deployed at each stakeholder's location when in use and when not in use. This provides a more economical and more affordable operational option. This reduces cost of deployment in several locations and cost of operational staff needed. This is presented in Figure 2 and Figure 3. Some applications can be made available on the middleware. These applications are accessible via web portals to connected stakeholders. This model is popularly known as Software as a Service (SaaS). The ability to access applications via web portals enables the use of a variety of devices. This includes PCs, tablets and internet enabled mobile phones. Mobile phones and tablets are cheaper devices that can be used in rural areas to access applications and interoperability services provided by the middleware. These devices are less expensive and rechargeable, making it advantageous for rural area that is characterized by shortage of finance and inadequate power supply to support e-Health services. These devices are also handy, mobile and easy to operate. This is depicted in Figure 4. This eliminates the need for sophisticated applications that require highly skilled manpower to operate, which is limited in availability in rural Africa.

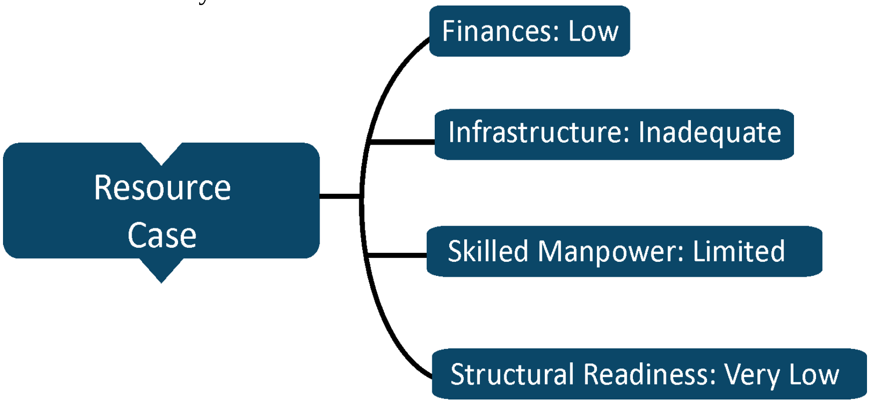
Resource Case
Finances: Low
Infrastructure: Inadequate
Skilled Manpower: Limited
Structural Readiness: Very Low

Figure 1: Resource Condition that Middleware must meet to be workable

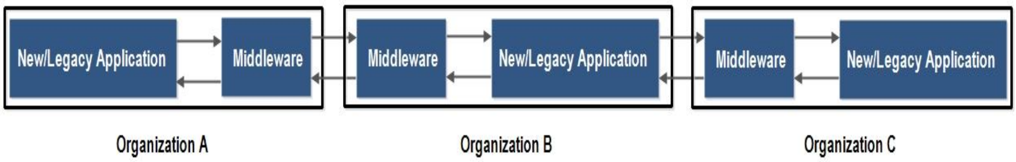

Figure 2: On Site Middleware Operational Mode

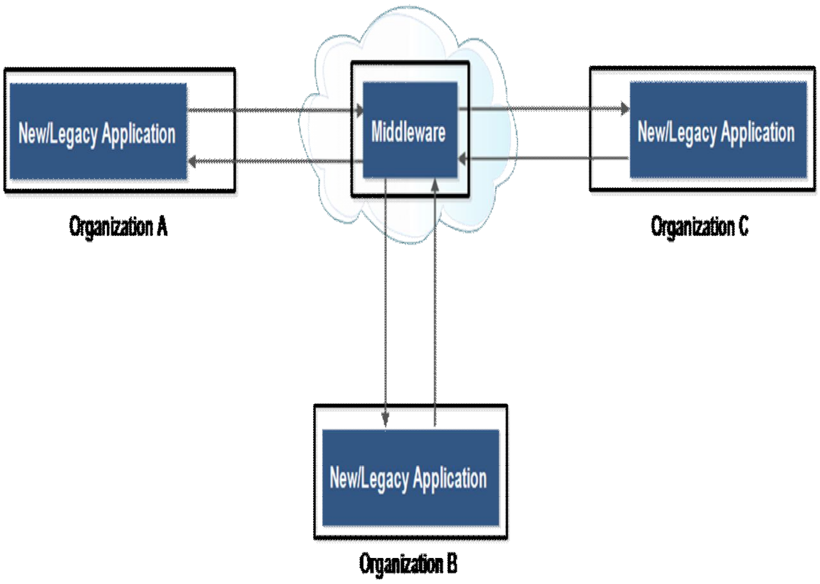

Figure 3: Platform as a Service (PaaS) Middleware Operational Mode

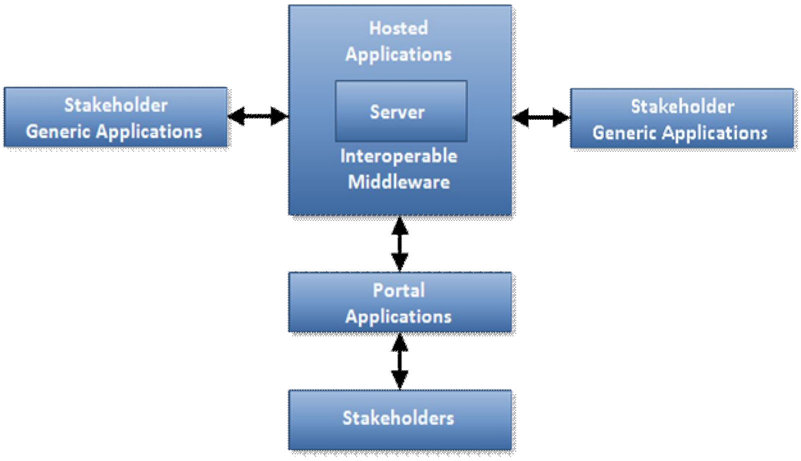

Figure 4: Software as a Service (SaaS) Middleware Application Mode

3. Middleware Design

The goal of the design is to provide a complete conceptual definition of the middleware parts to enable it meet the defined requirements. This results in designing the architecture of the middleware and supporting infrastructure in a layered approach. Since interoperability demands the integration of different e-Health applications as well as their different infrastructures, this results into an N: M relationship as each application still runs on different infrastructure. However, the individual e-Health service entities (application and infrastructure) interacts with the middleware in a M:1 relationship.

3.1. The Middleware Layer

The main goal of the middleware's architecture is to provide a platform for interoperability where other non-functional requirements are met. This includes reliability, maintainability, security and other stated non-functional requirements. The layered architecture consists of the following layer as shown in Figure 5.

i. The connected applications layer
ii. The interoperability middleware layer
iii. The supporting infrastructure layer

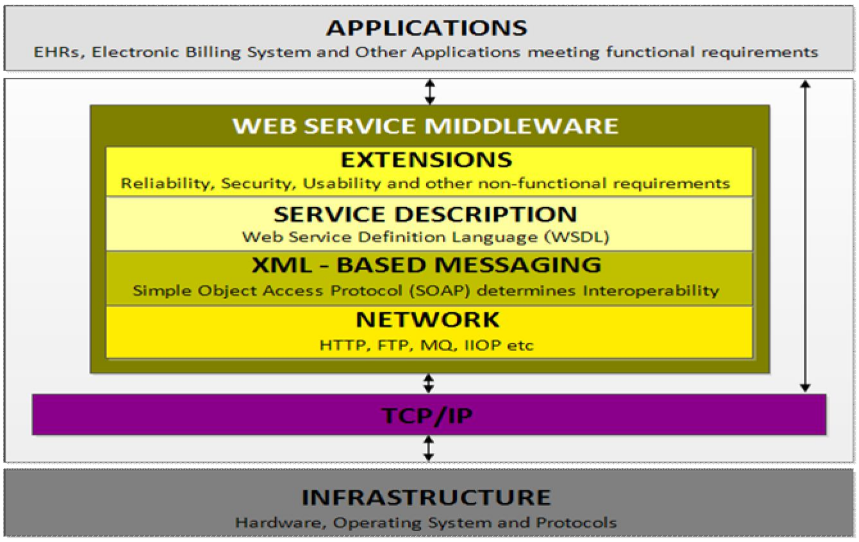
APPLICATIONS
EHRs, Electronic Billing System and Other Applications meeting functional requirements
WEB SERVICE MIDDLEWARE
EXTENSIONS
Reliability, Security, Usability and other non-functional requirements
SERVICE DESCRIPTION
Web Service Definition Language (WSDL)
XML - BASED MESSAGING
Simple Object Access Protocol (SOAP) determines Interoperability
NETWORK
HTTP, FTP, MQ, IIOP etc
TCP/IP
INFRASTRUCTURE
Hardware, Operating System and Protocols

Figure 5: Internal Architecture of the designed middleware

3.2. Survey of Existing Middleware Technologies

The following existing middleware technology were selected and evaluated with identified quality requirements.

i. Common Request Broker Architecture (CORBA)
ii. JAVA Remote Method Invocation (RMI)
iii. Distributed Component Object Model (DCOM)
iv. Secured Web Service

As a result of the comparative evaluation of the technologies, the Web Service middleware is selected as the middleware of choice and fare well in application-to-application integration. Web services are XML applications which can be used as interoperable building blocks to create more complex systems. Web service is a Message Oriented Middleware (MOM) currently emerging as the leading platform of choice for interoperability solutions in e-Health. Web Services contains four key technologies necessary for interoperability. These are:

i. **XML (eXtensible Markup Language):** This is a general markup language that can be used in a wide variety of contexts. Virtually all Web Service technologies make use of XML in one way or another.
ii. **SOAP (Simple Object Access Protocol):** This defines application-level interoperability. Its data representation is based on XML and is transported over HTTP.
iii. **WSDL (Web Service Definition Language):** This allows the specification of service interfaces.
iv. **UDDI (Universal Description, Discovery and Integration):** This serves the role of a mediator. Service providers and service requestors use the UDDI registry to establish connections to each other, this is also called binding.

3.3. Web Service Middleware Interoperability Operation

The main purpose of the middleware in design is to enable interoperability among different applications used by different stakeholders. The design of the interoperability operation consists of the following key concepts:

i. **XML:** This is an Extensible Markup Language format, abbreviated XML, and describes a class of data objects called XML documents. The representation of a document in XML is referred to as the schema. XML documents are made up of storage units called entities, which contain either parsed or unparsed data. Parsed data is made up of characters, some of which form character data, and some of which form markup. Markup encodes a description of the document's storage layout and logical structure.
ii. **Entity:** A combination of an e-Health applications and their respective infrastructure. Entities are heterogeneous due to their use of different standards and different infrastructure. An entity is used by a stakeholder. An entity is connected to a middleware, and can be requesting from another entity or providing information to another entity during the interoperability operation. An entity irrespective of its implementation standard that is capable of generating an XML document directly or processing it as an input can interact with the middleware directly; else it has to go through the XML Schema Library.
iii. **XML Schema Library:** Entities are implemented with different standards, yet interoperability is required across these different entities. The web service middleware utilizes the XML format for interoperability among different entities. However, not all entities can generate the XML format directly as output or process it as input. Hence, there is a need to have a XML schema library.

iv. **Service Provider:** A provider offers a service and wishes to advertise it for service requestors to use. In order to do so, the provider registers its service with the UDDI registry. The publication request includes information about the offered service, such as the WSDL specification.

v. **Service Requestor:** A service requestor is the entity looking for specific information from another entity. It does an inquiry to the UDDI registry, specifying what it is looking for which is described in the XML message. When there is a match, the UDDI registry responds with the information regarding a suitable service providing entity.

vi. **Service Registry:** This is implemented in UDDI. It is where service providers advertise and publish their services. This serves as a directory which service requesting entity can look into to locate a service providing entity. After this, a binding operation occurs between a service providing entity and service requesting entity.

It should however be noted that a single entity can be a Service Provider and Service Requestor at the same time. This is essentially dependent on the role each is occupying per time during the interoperability process. New EHR requesting information from an old EHR is a service requestor while the old EHR is the service provider. This can also be vice-versa. Figure 6 shows the activity diagram of the middleware Information Request Process. Figure 7 depicts the activity diagram of the middleware Information Acceptance Process. Figure 8 presents the sequence diagram of the middleware Interoperability Process.

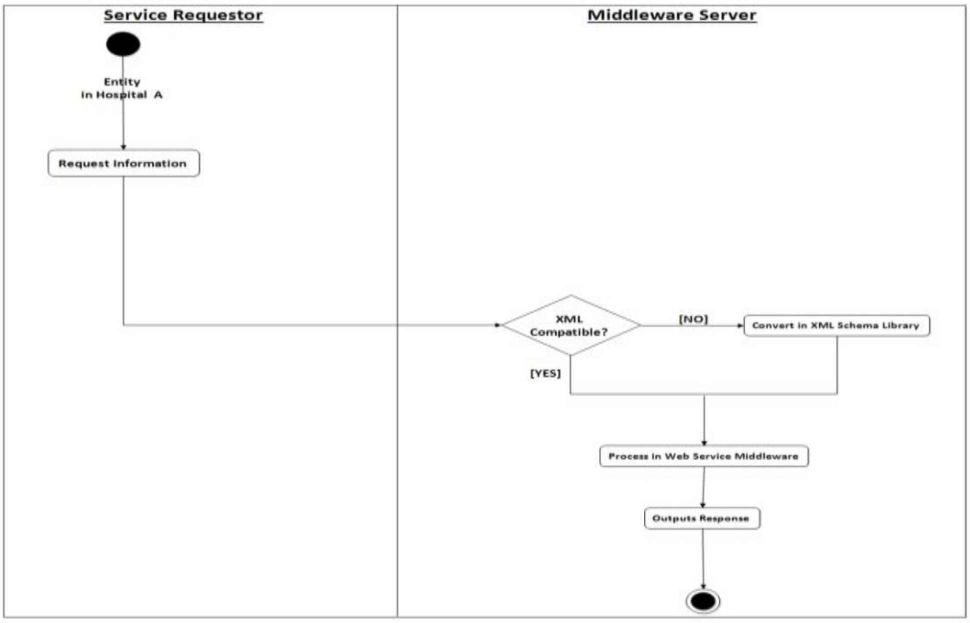

Figure 6: Activity diagram of the middleware Information Request Process

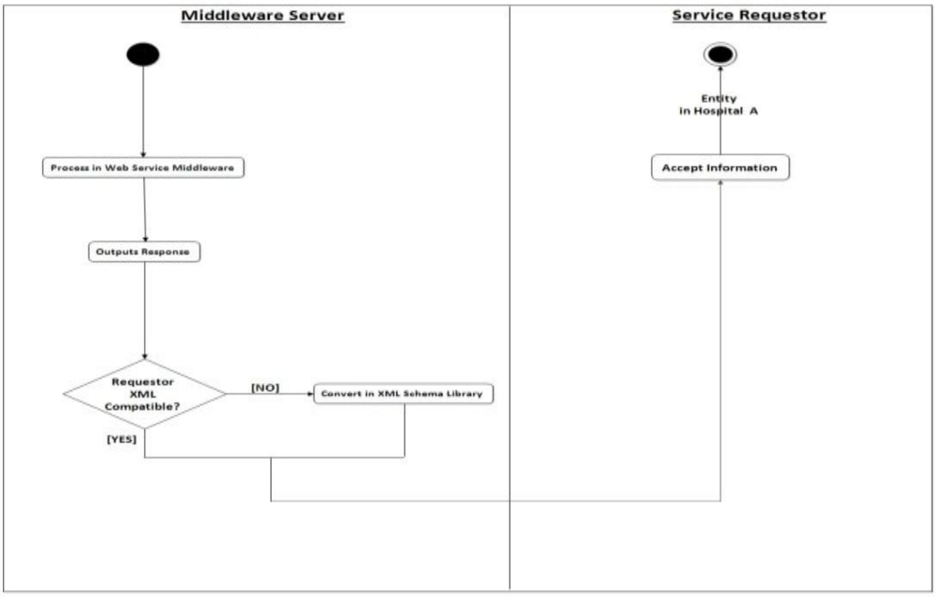

Figure 7: Activity diagram of the middleware Information Acceptance Process

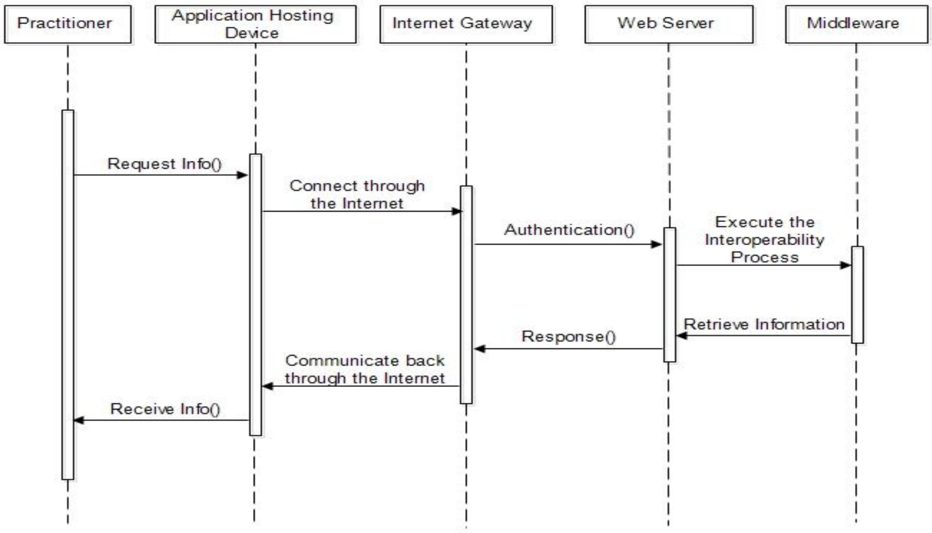
Practitioner
Application Hosting Device
Internet Gateway
Web Server
Middleware
Request Info()
Connect through the Internet
Authentication()
Execute the Interoperability Process
Retrieve Information
Response()
Communicate back through the Internet
Receive Info()

Figure 8: Sequence diagram of the middleware Interoperability Process

4. DEVICES AND ROLES

For both functional and non-functional requirements, the devices that can be interacting with the middleware are: i. Personal Computers, ii. Mobile Devices (smartphones and tablets), iii. Embedded Systems (medical devices/machines, tele-monitoring devices, server computer), iv. Data Storages and v. Network/Communication Devices (routers, cables, antennas, switches). Personal computers serve as an operational platform for applications connected to the middleware, same with mobile devices, though have less sophistication. Embedded systems could be information providers or requestors connected to the middleware. Data storages are used to store data and information during and after information exchange processes in the middleware. The network and communication devices support the infrastructure functionality of the infrastructure layer. These devices can be classified into four distinguishable roles, which incorporate the logically independent parts of the whole functionality of the middleware. These are: i. Client, ii. Server, iii. Registrar and iv. Mediator. Clients communicate and exchange information with the server. The Servers store information about clients and serve as database. The servers enable the information to be presentable in a XML style. To be operational in connection with the middleware, both client and servers must register at the registrar. The registrar provides a mechanism of authentication and authorization for the interoperability middleware. However, for legacy or non-compliant (to middleware standard format) entities to be operational, mediators are used. Mediators are gateways implemented as a XML Schema library of standards that can transform data structures from an incompatible manner to a compatible one.

5. MIDDLEWARE SECURITY AND PRIVACY

These are core quality requirements in the healthcare domain. The platform has several security features including an access control policy system for APIs, a multi-level authentication model, and encrypted communication channels. Authentication is available for entities that connect with the middleware. Entities are authenticated via signatures, and hosted web applications by HTTPS. Entities authenticate with different stakeholders having their own rights.

6. APPLICATION LAYER

The application layer comprised several e-Health applications that are connected to the middleware. The focus of this paper is on the Electronic Health Record (HER) and Medical Billing System (MBS) applications. The application layer also consists of a management tool for managing the middleware, with functionalities based on working middleware implementations. Each entity consisting of an application running on the infrastructure is connected to the middleware.

7. INFRASTRUCTURE LAYER

This design is not aiming at implementing another infrastructure technology but to have a design robust enough to integrate suitable infrastructure technologies. As for the network infrastructure, an All-IP solution is recommended to be used, by this strategy it was possible to natively integrate numerous devices, as long as they are able to speak IP and ensure that the requirements of middleware are met. Infrastructure technology consists of the software, hardware and network technologies required for the middleware to successfully execute the interoperability process. Figure 9 presents the infrastructure architecture of the middleware. Another area of concern is the internet connectivity option most suitable for rural e-Health services. The internet connectivity aids interoperability among different e-Healthcare services by facilitating their connection to the server, from which they access interoperability services. The three identified potential internet connectivity for the full scale interoperation of the interoperability service are:

i. Mobile Broadband (HSPA or more): This is an internet service provided by telecommunication companies primarily for mobile communication but can also be extended to other devices. The newer standard allows bit-rates to reach as high as 337 Mbit/s in the downlink and 34 Mbit/s in the uplink. 3G communication is an application of mobile broadband.

ii. Satellite Internet Access: is an internet service provider with a network based on wireless networking. Communication satellites provide internet service to ground based dishes which include VSAT.

iii. Power Line Communication: makes internet accessible via conductor lines that makes AC electricity transmission available to consumers.

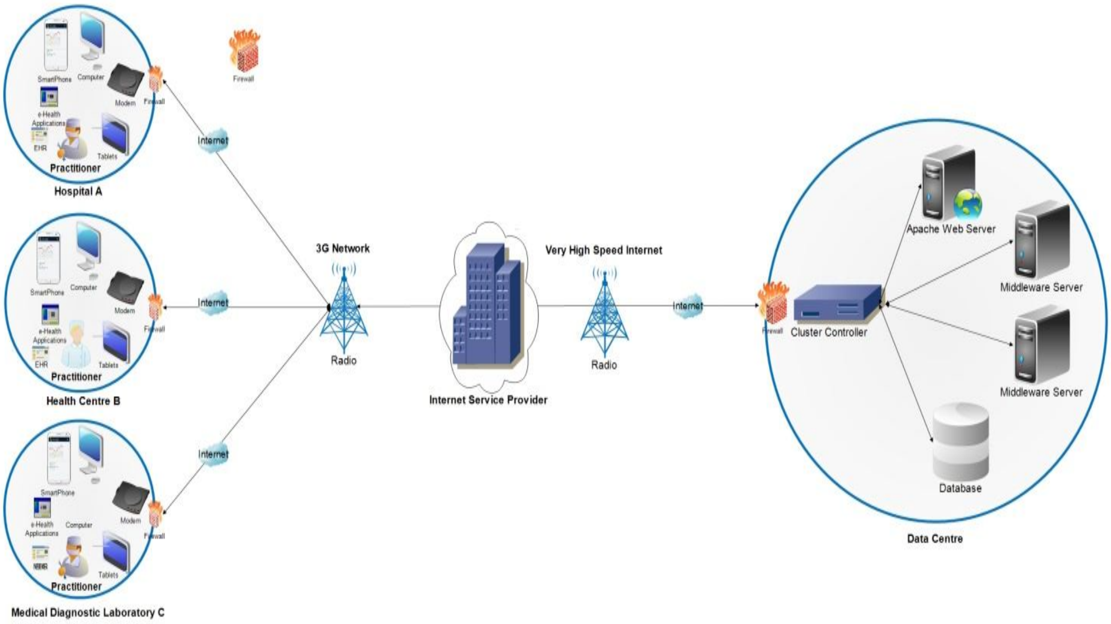

Figure 9: The Infrastructure Architecture of the Middleware

Mobile Broadband in form of GPRS, 3G communication and more is recommended as the internet connectivity option for implementation. This is due to its minimal cost and effort to set up. This is minimal due to the existing massive investment in telecom infrastructure by telecommunication companies in the country. Mobile broadband also offers convincing scalability properties as it can be extended with Wi-Fi access points. Rural areas where 3G network is not available can be easily catered for by the upgrade of existing telecom infrastructure in those areas. Mobile broadband though not readily available in all rural areas is recommended because it offers relative cost advantage to consumers. It has relative wider use compared to other internet connectivity options – use in telephony. However, in remote areas where mobile broadband access might be unrealistic, satellite internet service is better considered, though it is more costly and requires more technical expertise.

8. SYSTEM DEPLOYMENT

The designed middleware's deployment architecture is represented using a UML deployment diagram. It mainly consists of clients, servers and database (see Figure 10). These are further described as follows:

i. Clients

The clients are devices used in accessing and utilizing e-Health services. This could be a personal computer, medical device, tablet or smartphone. The e-Health applications can be generic or web based. Clients are the interfaces accessible to different e-Health stakeholders. The clients are connected to the servers via an internet network.

ii. Servers

These systems service the clients. The Apache web server services web or browser based applications. The middleware server services the web server and generic applications directly. The database is isolated from the middleware so as to maintain its integrity and reduce operation delays. The database serves as information source and sinks in the interoperability process. The cluster controller is used for load balancing and makes improvement in the efficiency of the interoperability operation by distributing the load demands across the different servers serviced by a single database.

iii. Databases

The database stores data required for the interoperability process. This covers that of various stakeholders and different organizations connected to the interoperability system. MySQL is an open server application that can be used to execute a number of queries.

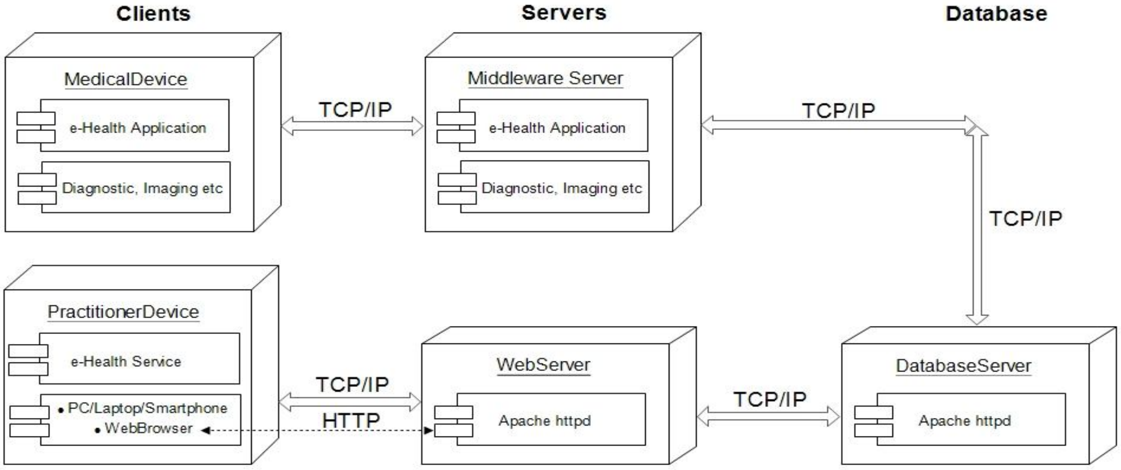

Figure 10: UML Deployment Scheme of the Interoperability Middleware

9. EVALUATION OF THE MIDDLEWARE

Making a selection over the identified products is difficult as a result of the complexity of the products. To make a selection of the appropriate middleware technology, each technology is mapped to the non-functional requirements that are identified. This mapping is mostly based on practical experience with these products and information found in reviewed literatures. The modifiability property of a system is mainly determined by the architectural styles used in its implementation.

9.1 Evaluation with Functional Requirements

Web Service is a general purpose middleware like the other middleware technologies. It can be used to realize distributed systems with any functional requirements. The satisfaction of the functional requirements is dependent on the system / application used by the user.

9.2 Evaluation with Non-Functional Requirements

This is a comparison of each considered middleware technology with the stated non- functional or quality requirements. These are: i. Interoperability, ii. Privacy, Security and Integrity, iii. Performance, iv. Maintainability, v. Development Time, vi. Testability. Tables 1 through 4 presents mapping of CORBA middleware to non-functional requirements, mapping of web service middleware to non-functional requirements, mapping of JAVA RMI middleware to non-functional requirements and mapping of DCOM middleware to non-functional requirements respectively.

Table 1: Mapping of CORBA Middleware to Non-Functional Requirements

CORBA	Quality Requirements
Interoperability	Platform Independent Language must have support for IDL binding
Privacy, Security and Integrity	CORBA security service supports a wide variety of features such as encryption, authentication, delegation, auditing, etc
Performance	High Performance
Maintainability	Becomes complex as it scales
Development Time	Complexity requires expertise to handle, and longer development time
Testability	Faults detectable by compile time type checking

Table 2: Mapping of Web Service Middleware to Non-Functional Requirements

Web Service	Quality Requirements
Interoperability	Platform Independent Any language is supported
Privacy, Security and Integrity	Web Services do not provide any standardized security services, although some aspects of security can be dealt

	with at transport protocol level. In particular, SOAP does not specify any security feature but rather makes it possible to exploit Internet technologies such as XML-Signatures or SSL to achieve maximum interoperability. This implies extended effort in comparison to other middleware technologies.
Performance	The conversion to and from XML results in a performance penalty and the extension into more features leads to a performance overhead
Maintainability	Ease of Maintenance with simplicity and extensibility as Interface is separated from implementation.
Development Time	Shortened development time as services can be easily published and subscribed to.
Testability	Lack of compile time checking leading to harder debugging, but code is very easy to read

Table 3: Mapping of JAVA RMI Middleware to Non-Functional Requirements

JAVA RMI	**Quality Requirements**
Interoperability	Primarily suited to Java Virtual Machines, but implementations now exist for non-JVM machines
Privacy, Security and Integrity	Security provided by default and customizable
Performance	Performance highly dependent on data type and requires more data
Maintainability	Easier to set up than CORBA
Development Time	Integration of non-Java products and time consuming
Testability	Faults detectable by compile time type checking

Table 4: Mapping of DCOM Middleware to Non-Functional Requirements

DCOM	**Quality Requirements**
Interoperability	Mainly Windows, but it can support other platforms with some efforts. Language must have IDL binding.
Privacy, Security and Integrity	DCOM security is tightly integrated with Windows NT security. This is not advantageous for non-Microsoft platforms.
Performance	Extensible for the optimization of performance
Maintainability	No longer promoted by Microsoft, support might be difficult to obtain.
Development Time	Extension to non-Microsoft platform can be time consuming.
Testability	Faults detectable by compile time type checking.

The mapping of the selected middleware technology with the identified quality requirements reveal that web services still have the highest support for interoperability. This goes beyond the operating platform to the language itself. Interoperability being the primary goal of the designed middleware is best met by the web service middleware. The simplicity and extensibility of the language also makes it very easy to maintain as services can be added or removed easily via the publishing and subscription mechanism. This is made possible by the ability to separate the implementation from the interface. The simplicity makes it easy to maintain when needed without getting into too much complexity as seen in other distributed middleware technologies. It should also be noted that in this internet dominated age and the advent of the semantic web, web service offers the most internet friendly middleware interoperability service. However, web services tend to lag behind than other middleware technologies like DCOM and CORBA in performance. This is due to its Message Oriented Messaging nature, where more overheads are required in making conversions to XML and transmissions from one point to another. This problem can be solved by making more investment in the acquisition of processing units and servers which offers more Return On Investments (ROI) over the long term than hiring a team of personnel to handle more complex middleware technologies which gets more complex than they scale. Another pitfall of web service is the inability to check type at compile time. This makes debugging difficult. This problem is being solved by the increasing availability of third party tools such as SOAPUI and Apache Axis tools.

10. CONCLUSION

This work provide an insight into accessing middleware via web portal using variety of devices which includes PCs, tablets and internet enabled mobile phones which are cheaper devices that can be used in rural areas to access applications and interoperability services provided by the middleware. The interoperability middleware design work took into consideration the deployment environment requirements through usage of devices that are less expensive and rechargeable, making it advantageous for rural area that is characterized by shortage of finance and inadequate power supply to support e-Healthcare services. These devices are also handy, mobile and easy to operate. This eliminates the need for sophisticated applications that require highly skilled manpower to operate, which is limited in availability in rural Africa.

REFERENCES

[1] Clark, T. (2007). Adopting Healthcare Informatics and Technologies. Americal Journal of Health System Pharmacy, vol. 64.

[2] Alvarez, C.R. (2002). The promise of e-Health- a Canadian perspective. Journal of e-health Int., licensee BioMed Central Ltd., doi:10.1186/1476-3591-1-4.

[3] Katehakis, D.G., Tsiknakis, M. and Orphanoudakis, S.C., (2002). A healthcare information infrastructure to support integrated services over regional health telematics networks. Health IT Advisory Report, 4(1), 15-18.

[4] Van, B.J., Musen, M.A., Houten, A.W. and Verlag, S. (1997). Handbook of Medical Informatics.

[5] Lyytinen, K. and Hirschheim, R. (1987). Information system failure - a survey and classification of empirical literature, Oxford Surveys in Information Technology, 4:257-309.

[6] Bernstein, M.L., Mccreless, T. and Cote, M.J. (2007). Five Constants of Information Technology Adoption in Healthcare, Winter Journal, vol. 85, no. 1.

[7] Denise, S. (2003). The case for e-health, Available at: http://www.eipa.nl/Publications/Summaries/03/2003_E_01.pdf.

[8] Hanson, C.W. (2006). Healthcare Informatics. McGraw-Hill, New York, NY. ISBN 0-07-144066-6. 227 pages.

[9] Anthony, S. C., Bensik, M., Armfield, N., Stillman, J. and Caffery, L. (2005). Telemedicine and rural healthcare applications. Journal of Postgraduate Medicine, 51(4), Oct-Dec: 286-293.

[10] Maheu, M.M., Whitten, P. and Allen, (2001). A. E-Health, telehealth and telemedicine: A guide to start up and success.

[11] Doolittle, G. C., Williams, A., and Cook, D. J. (2001). An estimation of costs for a pediatric telemedicine practice: Results from the TeleKidcare experience.

[12] Jennett, P., Healy, T., Jackson A., Woollard, R., Ho, K., and Bates, J. (2001) Framework for Remote and Rural Readiness in Telehealth - First Steps. The Future of Health Care Proceedings on CD COACH Conference. Toronto, Ontari.

[13] Lundh, E. and Sandberg, M. (2002), Time constraint requirements engineering with extreme programming –an experience report, IEEE Joint International Requirements Engineering Conference, Essen, Germany.

[14] Carr, J. (2000), Requirements engineering and management: the key to designing quality complex systems, TQM Magazine, Vol 12, No 6, 400-407, MCB University Press, ISSN 0954-478X.

[15] Terwijin, B., (2005). Selecting Middleware for the Intelligent Autonomous Systems Group,