PERINTIS eJournal

PERINTIS eJournal, **2020**, Vol. 10, No. 1, pp. 25-33

MOBILE VIEWER SYSTEM DESIGN FOR EXTENDING MEDICAL DIAGNOSTIC IMAGING NETWORKS

Azi Azwady Jamaludin[1], Azizul Azizan[1*], Hafiza Abas[1], Salwani Mohd Daud[1], Yusnaidi Md Yusuf [1], Hadhrami Ab. Ghani[2]

[1] Razak Faculty of Technology and Advanced Informatics, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia.

[2] Institute for Artificial Intelligence and Big Data, Universiti Malaysia Kelantan, Kelantan, Malaysia.

[*] Corresponding author: azizulazizan@utm.my

ABSTRACT

Medical diagnostic imaging or commonly known as DICOM has been developed to meet the needs of manufacturers and users of medical imaging equipment for data interconnection between devices and modality (i.e. medical imaging) on standard networks. The medical data and image that is captured as DICOM data attributes, can be transmitted and processed between various DICOM devices and software, hence limiting the access of DICOM data to non-standardized DICOM systems. Herein, an internet based Picture Archiving and Communication System (PACS) server with web viewer with the capability to extend the DICOM network through network gateway is demonstrated. It enables the web viewer to display the DICOM image using plugin, directly from DICOM archives. Further functionality of the server has enabled the implementation of relational database plugin and the stored source can be shared by using normal mobile application. This paper discusses the system design of stand-alone mobile viewer for DICOM, based on web service with relational database and an internet based PACS server based on client server architecture. The outcome of the study enables displays of medical images within a non-DICOM networks via mobile web-based application.

KEYWORDS: *Medical imaging, DICOM modality, PACS server, Relational database, Web service, DICOM viewer.*

1.0 INTRODUCTION TO DICOM STANDARD

The DICOM standard which stands for Digital Imaging and Communication in Medicine was introduced for point to point connection in order to achieve wider access of image sharing within the medical facilities network. The history of DICOM protocol starts since 1983 when it was introduced by American College of Radiology (ACR) and National Electric Manufactures Association (NEMA). Before it was known as DICOM, the first version of the protocol was named as ACR-NEMA 300-1985, and the second version was renamed as ACR-NEMA 300-1988 [1]. DICOM images and its metadata is stored in a Picture Archiving and Communication System (PACS) server for repository.

A DICOM file has two main categories of services i.e. the composite and normalized services. Bidgood et al. [3], stated that DICOM record updating is disabled for composite services, where any amendment to the record or images issued requires an issuance of a new DICOM file. In contrast, normalized services support creation, updating and deletion of records but these updates are limited to certain datatype. The images can be managed by using specific commands based on the services, where the initial C character such as C-STORE represent composite services and initial N character such as N-SET represent normalize services. These applied command does not require a dedicated application server for image distributions.

Horii et al. [2], has noted that the DICOM network for image sharing is very much limited within DICOM network ecosystem. Just recently, the latest DICOM protocol has been devised to support common internet and web technology; enabling the DICOM web-based viewer to be built besides a stand-alone DICOM viewer. A study by [4] stated that the new DICOM network environment enables a DICOM file to be transferred using Upper Level Protocol (ULP) over Transmission Control Protocol (TCP), and Internet Protocol (IP), where TCP ports of 104, 2761, 2762 and 11112 is usually used.

2.0 DICOM IMAGING DATA AND VIEWER

Diagnostic devices and modality produce voluminous medical data files pertaining to the patient diagnosis and patient record. The DICOM files according to Golubev et al. [5], is usually created in series based on successive cross sections of the corresponding body organ. The DICOM file represent an animation of multiple images slices where fast rendering of the images can produce a motion picture well suited for 3-dimensional (3D) diagnosis. Meenatchi et al. [6] stated that DICOM images need to be shared between healthcare institutions to reduce patient radiation exposure due to image recapturing.

Traditionally, the captured DICOM images is written or stored into compact disc (CD) for sharing and distribution. Even though the Internet enables a convenient way of sharing DICOM object, there is limitations in terms of images access. According to Liang et al. [7], when the DICOM image access is being broaden via web technology, many PACS server manufacturers are using medical image reading system based on Browser Server (B/S) architecture i.e. Internet based PACS server equipped with DICOM web viewer. Although the DICOM web viewer enables broader access to DICOM images by fulfilling web-based requests from other client, the image quality displayed is lower compared to stand-alone DICOM viewer.

Dedicated stand-alone DICOM viewer applications such as OsiriX utilize its own plugins to communicate directly to DICOM equipment hence giving much better image quality.

Previously, OsiriX application cannot be deployed outside the DICOM network, but recent improvements have enabled the OsiriX to be integrated with recent internet based PACS server. This enables the DICOM objects to be retrieved from the cloud.

2.1 DICOM Web Viewer Implementation

DICOM viewers can be developed as a web viewer or stand-alone viewer and can be integrated to cloud or confined within an internal network devoid of external access. The initial development on web viewer is usually based on Java applet. Ellerweg et al. [8] stated that web-based DICOM viewer can also be implemented by the X toolkit, which is a Web Graphics Library (WebGL) for rendering interactive 3D computer graphics. The DICOM network architecture according to Ellerweg et al. [8], is managed in two sections. The first section is the image repository section that stores the DICOM objects and is typically implemented in a healthcare closed DICOM network. Meanwhile for the web-based section, any web browser can access the DICOM objects from anywhere as long as there is an Internet access.

Web technologies such as Virtual Reality Modeling Language (VRML) for 3D display have been used. However, when VRML became obsolete DICOM web viewer started to implement web viewer based on Flash multimedia animations [9]. Mobile device is equipped with web browser, but the experience is not the same compared to desktop browser due to lower screen resolution and are unable to support certain technology available in desktop-based viewers. Moreover, Flash animations is not supported by mobile web browser.

2.2 Cloud based data retrieval

The data within a cloud environment is seldom processed with encryption thus making confidential patient data deployment in cloud computing to be less secured. Doukas et al. [10], mentioned that cloud service is well suited for business continuity or for backup i.e. business data recovery. Nevertheless, the main drawback of cloud computing is lack of security. Cloud storage represents a virtual folder and does not require database query features for data retrieval and access. To access database locally, such as SQLite, database queries must be executed using database commands that is embedded together with stand-alone application.

Data retrieval can be performed through Representational State Transfer Application Programming Interface (REST API) or RESTful web service which is similar to services provided by Amazon Web Service (AWS) cloud. According to Narula et al. [11], the security in AWS cloud is achieved through firewall, traffic flow policies and access control list. The AWS cloud also has a limited number of access point where its connection can be accessed through Hyper Text Transfer Protocol Security (HTTPS). Furthermore, the service is segregated from Amazon production network where the access to the service is enabled through a ticketing system.

2.3 DICOM data retrieval: Cloud versus Internal

The main difference of data access between an internal network data repository and cloud repository is the storage architecture and component. Cloud computing refers to storing and accessing data services over the Internet instead of the traditional local storage devices. Cloud facilities communicate with readily available RESTful web service and synchronizes the virtual folder with local folder. Hence, the retrieved data can be queried by application embedded database such as SQLite.

For DICOM data retrieval, a wider access to DICOM objects is materialized with the support of DICOM protocol that supports the Internet and web technology. The web viewer enables users using a web browser to display the image and data, with a significantly lower image display quality than a stand-alone DICOM viewer application. To note, the current functionality of stand-alone viewer has enabled DICOM objects to be displayed in a non-DICOM network. This is achieved through integration of cloud computing and integration with an Internet based PACS server.

From our findings, the configuration of DICOM data archiver has made it hard to enable the direct DICOM database integration to cloud facility. In practice, the medical staff needs to upload the DICOM images and data manually to cloud storage. Recently, internet based PACS such as Orthanc server as elaborated by Jodogne [12], and Oladiran et al., [14]. has shown that it can support direct integration to DICOM archiver or DICOM modality. The objective of preserving the exclusiveness of DICOM meanwhile providing a wider access to non DICOM stand-alone viewer can be achieved by utilizing the RESTful feature of Orthanc server. Herein, the load of the Orthanc server can be further reduced if a separate RESTful web service is developed to access the DICOM database in an Orthanc server.

3.0 METHODOLOGY

This mobile viewer and web service application can be deployed as stand-alone application located at different sides of the network. The mobile viewer lies at the client side whilst the web service lies at the server side. This architecture is known as client server architecture. The stand-alone mobile application is developed using native Software Development Kit (SDK) whilst RESTful web service is developed using Java Development Kit (JDK). The Orthanc server is deployed as an Internet based PACS server able to generate a database table. The DICOM image and its embedded text data is stored as a Binary Large OBject (BLOB) data type before the text data is extracted to tag the data according to the DICOM standard.

There are currently two famous mobile platform that is iPhone Operating System (iOS) and Android Operating System (Android OS). The iOS platform uses Swift library as the specific SDK whilst Android OS uses Kotlin SDK [13]. Application developed in JDK is usually a cross platform, but an application using graphical interface libraries such as Standard

Widget Toolkit (SWT) library compiled with JDK application is a platform dependent. Herein, the RESTful JDK based web service is not using any interface library where its main function is to focus on client server integration.

Basically, the implementation of Orthanc server helps in achieving the wider access for DICOM data retrieval. It is an alternative to upload the DICOM data to the cloud storage where it automatically run a generated relational database script and share the database with other DICOM network independent application. Furthermore, Orthanc server comes with built in web viewer and can easily integrates to DICOM dependence viewer application such as OsiriX using AET configuration.

4.0 RESULTS AND DISCUSSION

The objective of this mobile viewer development is to display the DICOM image and data on mobile device that is independent and external from a DICOM network but at the same time maintains the exclusiveness of the DICOM object. To materialize this objective, DICOM image and data must enable to be uploaded to TCP compatible storage or relational database where Sequence Query Language (SQL) can be performed.

The database scheme is configured with default sorting rule, character, username and password. The username, password, scheme name and database port are then used in Orthanc server configuration file. For MySQL plugin, an extra socket path is also added into the configuration file. Once the server is running, it automatically creates a database table and datatype for each column.

4.1 Network Topology

DICOM viewer application such as OsiriX is much dependent on DICOM network because it implements its own protocol compatible to previous DICOM standard i.e. the ACR-NEMA protocol. Moreover, such stand-alone cloud application interacts with DICOM files that is uploaded manually to cloud facilities by utilizing REST API web service.

Herein, an Orthanc server is implemented for our system design to establish the communication with DICOM archiver on DICOM networks. The Orthanc server utilizes Application Entity Title (AET) and retrieve the DICOM data using Web Access to DICOM Object (WADO) to enable DICOMweb plugins. Orthanc server is compatible with TCP protocol thus it can interact with any relational database and any web-based browser to display the DICOM image from the database after converting the images to JPEG format. Figure 1 represent the proposed DICOM distribution system through an Orthanc server to extending the DICOM network outside of conventional DICOM network.

PERINTIS eJournal, **2020**, Vol. 10, No. 1, pp. 25-33

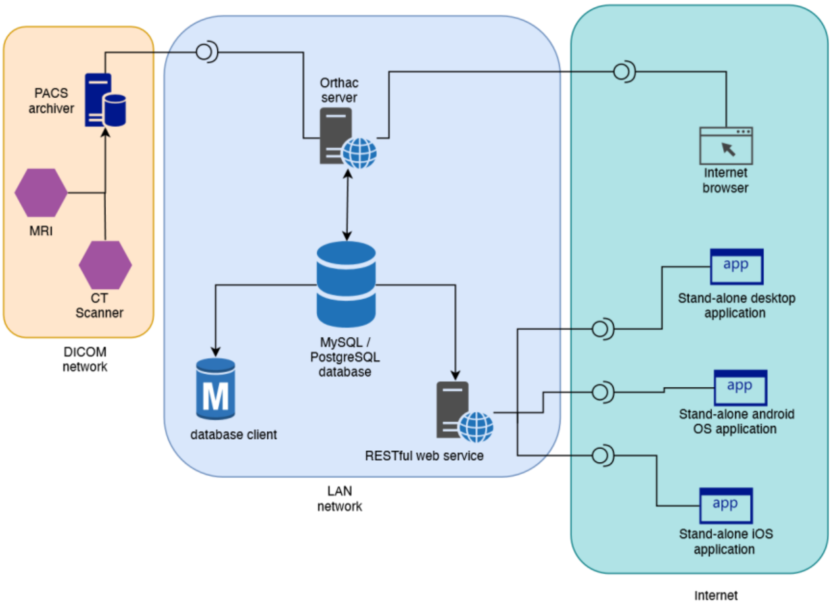

Figure 1 DICOM distribution through Orthanc server

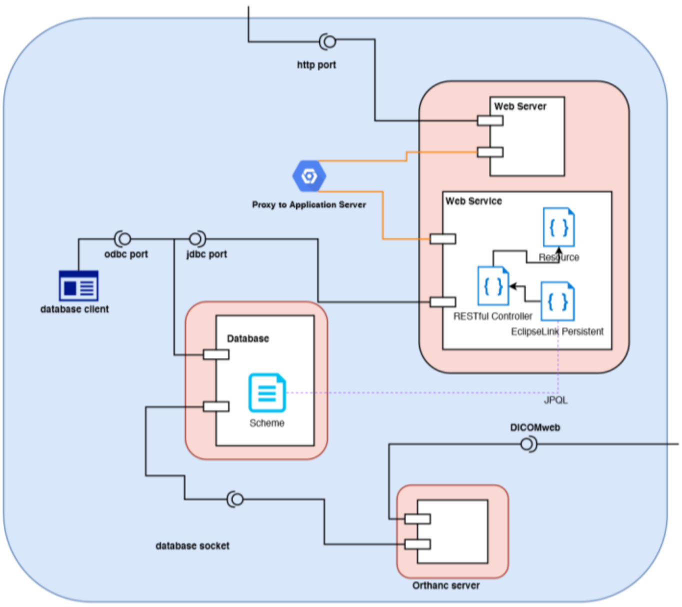

Figure 2 LAN network connectivity

4.2 Database Interaction

In Figure 1, the retrieved DICOM data is uploaded into a configured MySQL or PostgreSQL database. For database that is not configured, Orthanc can use its available embedded SQLite database. The configured database can also interact with other application such as database client. Database administrator can implement the Command Line Interface (CLI) or run the database client such as MySQLWorkBench to interact with database using Graphical User Interface (GUI). Once the connection has been established, it can receive a query for data retrieval from the application via the Internet.

4.3 Service Artifact

Figure 2 details the Local Area Network (LAN) portion. The communication to database is established through connectivity ports such as Java Database Connectivity (JDBC). Herein the persistent artifact retrieves the data stored in database scheme using the query language of Java Persistent Query Language (JPQL). The figure also shows, stand-alone application such as mobile application which is located outside the LAN network. These applications are able to communicate with the web server by sending request through http port and web server. The request is forwarded through a web server proxy to the RESTful web service. Carmatec [15] list out the difference between web service and web application lies in the GUI where web service does not necessarily have any GUI. It is also noted that web services is also independent of any programming languages or platform, where Java based RESTful or similar client server services can be accessed by platform specific application such as application developed in Microsoft Windows. By implementing this solution, the web service can also be used by other web application for data request.

5.0 CONCLUSIONS

Most networks implement a gateway as a tool to extend data access from the existing network. This approach is similarly adopted for the DICOM networks by implementing a specialize gateway using the Orthanc servers as discussed above. Furthermore, the new DICOM protocols has enabled the possibility of developing an internet based PACS server to distribute DICOM files and objects for non DICOM networks. This client server architecture establishes the communication protocols to the DICOM archiver through AET, and TCP/IP. Furthermore, it can also be configured with a relational database to be used for mobile data access using web-based viewer applications.

Acknowledgement

Special thanks to Orthanc developer for utilizing TCP/IP protocols thus simplifying the DICOM distribution to non DICOM networks. This materializes the concept of sharing the DICOM file with other networks, while maintaining the exclusiveness and integrity of DICOM object that is critical for medical data.

REFERENCES

[1] Pianykh, O.S. (2012) What is DICOM?. In: Digital Imaging and Communications in Medicine (DICOM). Springer, Berlin, Heidelberg.

[2] Horii, S.C. Prior, F.W. Bridgood, W.D. Parisot, C. Claeys, G. (2018). DICOM: An introduction to the standard. https://www.csd.uoc.gr/~hy544/mini_projects/Project8/DICOM%20(Paper_Parisot).doc. (Accessed on 30 October 2019)

[3] Bidgood, W. D., Horii, S. C., Prior, F. W., & Van Syckle, D. E. (1997). Understanding and Using DICOM, the Data Interchange Standard for Biomedical Imaging. *Journal of the American Medical Informatics Association*, 4(3), 199–212.

[4] Datar, K. (2016). An Introduction to DICOM (Digital Imaging and Communications in Medicine). ExtraHop. https://www.extrahop.com/company/blog/2016/introduction-to-dicom-protocol/. (Accessed on 29 October 2019)

[5] Golubev, A., Iliuha, N., & Bogatencov, P. (2017). DICOM Network services DICOM data exchange solution integrated in the regional VI-SEEM infrastructure. 17th IEEE International Conference on Smart Technologies, EUROCON 2017 - Conference Proceedings, (February 2015), 558–563.

[6] Meenatchi Aparna, R. R., & Shanmugavadivu, P. (2017). A literature review on medical imaging transfer. Proceedings of the International Conference on IoT in Social, Mobile, Analytics and Cloud, I-SMAC 2017, 284–288.

[7] Liang, B. J., & Lin, Y. J. (2017). A web-based mobile medical image reading system. Proceedings of the 2016 8th International Conference on Information Technology in Medicine and Education, ITME 2016, 50–53.

[8] Ellerweg, R., Reuter, D., & Weir, P. (2016). Architecture of a web-based DICOM viewer showing segmentations and simulations. 2016 IEEE 18th International Conference on E-Health Networking, Applications and Services, Healthcom 2016, 0–4.

[9] Melicio Monteiro, E. J., Costa, C., & Oliveira, J. L. (2013). A DICOM viewer based on web technology. 2013 IEEE 15th International Conference on E-Health Networking, Applications and Services, Healthcom 2013, (Healthcom), 167–171.

[10] Doukas, C., Pliakas, T., & Maglogiannis, I. (2010). Mobile healthcare information management utilizing Cloud Computing and Android OS. 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10, 1037–1040.

[11] Narula, S., Jain, A., & Prachi. (2015). Cloud computing security: Amazon web service. International Conference on Advanced Computing and Communication Technologies, ACCT, 2015-April, 501–505.

[12] Jodogne, S. (2018). The Orthanc Ecosystem for Medical Imaging. Journal of Digital Imaging, 31(3), 341–352.

[13] Saccomani, P. (2018). Native, Web or Hybrid Apps? What's The Difference?. https://www.mobiloud.com/blog/native-web-or-hybridapps/. (Accessed on 30 October 2019).

[14] Oladiran, O., Gichoya, J. and Purkayastha, S., (2017). Conversion of JPG Image into DICOM Image Format with One Click Tagging. In International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management, 61-70

[15] Carmatec. (2018). Web Services vs Web Applications Carmatec Inc. https://www.carmatec.com/blog/web-services-vs-webapplications/ (Accessed on 30 October 2019)