International Journal of Database Theory and Application
Vol.9, No.7 (2016), pp.281-290
http://dx.doi.org/10.14257/ijdta.2016.9.7.24

A Structured and Layered Approach for a Modular Electronic Voting System: Defining the Application Layer

Emeka Reginald Nwogu[1] and Chinedum E. Ihedigbo[2]

[1]*Michael Okpara University of Agriculture, Umudike, Nigeria*
[2]*Computer Science Department*
Michael Okpara University of Agriculture, Umudike, Nigeria
okeynedum@yahoo.com, nwoguemeka@gmail.com

Abstract

This work is the first of a series of works that discussed a model for solving the problem of non-modularity in electronic voting systems. It analyzed and described the system from a structured and layered perspective; with the system layered in order to achieve modularity. First, was the description of similar models and architectures that have been previously proposed by other authors on related subjects and how good their models have been. Our proposed system was structured into three layers that include Application, Security Service and Network Access layers. However, in this work, focus was more on describing the Application layer, which is further split into two sub-layers, namely the Application Hardware and Application Software sub - layers. A couple of components and modules that make up this layer were carefully outlined, reviewed and discussed.

Keywords*: E voting, Layer, Modular, Application, Security Service, Network Access*

1. Introduction

Electronic voting has recently enjoyed more popularity and publicity, especially as many nations are now exploring the possibility of owning and deploying such system. This system makes the organization and execution of national elections somewhat hitch free. Citing [11], electronic voting system (also known as e – voting) is an electronic system which uses electronic ballot that would allow voters to transmit their secure and secret voted ballot to election officials over the computer. [18] also in their discussion of an electronic voting (e-voting) system posited that the system is one in which the election data is recorded, stored and processed primarily as digital information. Lastly, [13] described e-voting system as one that allows the eligible voter to cast their vote via a computer normally connected to the internet or intranet from any location.

Though electronic voting has evolved over the years in their designs, starting from the days of the punched cards to the present day Direct Recording Electronic (DRE) voting system; it has continued to suffer lack of standardization and protocol definition. This unarguably, has made it impossible for system designers and manufacturers to pursue a generic viewpoint to e –voting system design. As [1] puts it, it appears that each researcher in the field of Electronic Voting Systems contributes to some particular aspect but rebuilds the whole system when they wish to implement this rather specific contribution. This issue has made the e –voting systems un-maintainable and un-scalable

Similarly, the most disturbing issue with most of the systems that have been developed and deployed so far has been the issue of non-modularity in these systems. According to [19], Modularity is the degree to which a system's components may be separated and recombined. [12] defined modular products as systems of components that are loosely coupled. Loose-coupling on the other hand has been defined in computing and systems design as one in which each of its components has, or makes use of, little or no

ISSN: 2005-4270 IJDTA

knowledge of the definitions of other separate components. Non-modularity has made it difficult for the electronic voting systems or modules to be compatible or integrate with modules from other systems. Modularity cannot be achieved without layering electronic voting systems and defining the protocols and modules in each layer of the system.

This work attempts to solve the problem of non-modularity by analyzing and conceiving an electronic voting system from a structured and layered perspective and viewpoint. The work further defined the different modules, components and protocols that sit in each layer.

2. Layering Approach in Information Systems

It is difficult to change one element of an information system without affecting the others. This normally gives rise to system level maintenance as against component or module level maintenance. The goal of layering is precisely to offer an intelligent way to segment infrastructure so as to allow for changes in the information system while ensuring easy installation. According to [7], layering in computer programming is the organization of programming into separate functional components that interact in some sequential and hierarchical way, with each layer usually having an interface only to the layer above it and the layer below it. [10] also defined layering today to involve segmenting an information system into modular, interdependent layers; where each layer is then minimally tied to the other layers.

The first successful attempt at introducing layering in information systems in order to improve performance and compatibility was the TCP/IP model created by the United States Department of Defense (DoD) in the early 1970s. This was a sequel to the first Request for Comments (RFC) published in April 1969, which paved way for today's Internet and its protocols [16]. TCP/IP provides end-to-end data communication specifying how data should be packetized, addressed, transmitted, routed and received [14]. This functionality is organized into four abstraction layers which are used to sort all related protocols according to the scope of networking involved [15]. This model enabled compatibility and inter-operability between devices from different companies. The introduction of the DoD's TCP/IP model led to a similar model from International Organization for Standardization (ISO) which created the Open System Interconnection (OSI) model in 1984.

Similarly, the concept of layering has been applied in computer and system security. [5] and [3] believe that a layered approach to security provides better protection of Information Technology systems.

3. Related Works

A couple of writers and technology analysts in electronic voting systems have tried to design the system in different ways. [17] in their electronic voting architecture suggested a tier based approach to electronic voting, with client, Application server and the Database server tiers. However, a tier based approach to electronic voting does not discuss the e voting system comprehensively. Rather, it looks at the system only from the application or software perspective.

Similarly, [6] designed his electronic voting architecture based on the Glue Meta model. He further suggested a three-layer architecture that includes the Voting Machine Layer, Glue and Gates Layer and Central Servers Layer. His Voting Machine Layer comprised of client Machines that record votes cast by the voters at the voting centers (polling booth). Similarly, his Glue and Gates Layer comprise of the Glue which he called the big ballot store where every Voting Machine sends the ballots recorded. And the Gate which he said allows communication between Voting Machines and Glue. Also, his Counting Servers Layer comprised of the trusted Central Server Machine which counts and collates the results of the election. This approach did not discuss the electronic voting

system network infrastructure and security requirements. Consequently, it did not solve the problem of non-modularity in electronic voting systems.

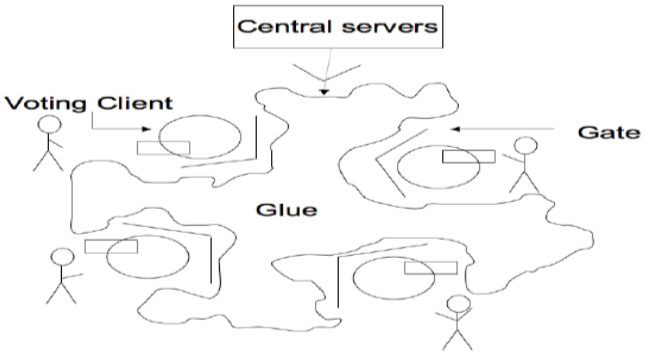

Figure 1. Glue Architecture for Electronic Voting as Suggested by [6]

Apparently, the closest attempt at solving this problem was from [1]; but his described layers did not capture the whole infrastructure and resource requirement of the electronic voting system. Though, he shared the same viewpoint of designing a modular electronic voting system by layering the system, his suggested layers were not very comprehensive. The system in [1] comprised of four layers with each layer having several components.

Table 1. Layering of the Electronic Voting System as Proposed by [1]

Layer	Components
Human layer	Voter registration Ballot form configuration Polling station layout and management Verifiability front-end
Election layer	Election method Election management system Voter-ballot box communication channel
Computational layer	Cryptography scheme Anonymisation strategy Tallying procedure
Physical layer	Hardware authentication method Publishing strategy Transfer method

4. The Proposed Model

There is need for a layering structure that encompasses the infrastructure, the platform, the software and all services as part of the complete system. The system presented here attempts to layer the electronic voting system by building all the necessary resources including infrastructure, platform, software and services into the specification.

Our proposed system has three layers which include, the Application Layer, Security Service Layer and Network Access Layer. Information and control must not necessarily flow down sequentially from the Application layer down to the Network Access Layer. There is a horizontal association between the Application Layer and the Security Service layer. Consequently, information may flow between the components of this pair severally before moving down to the Network Layer.

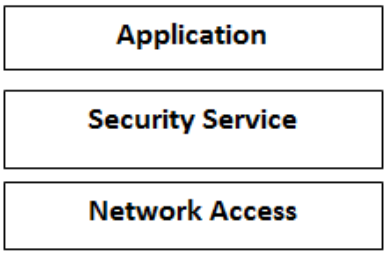

Figure 2. Three Layer Voting System Model

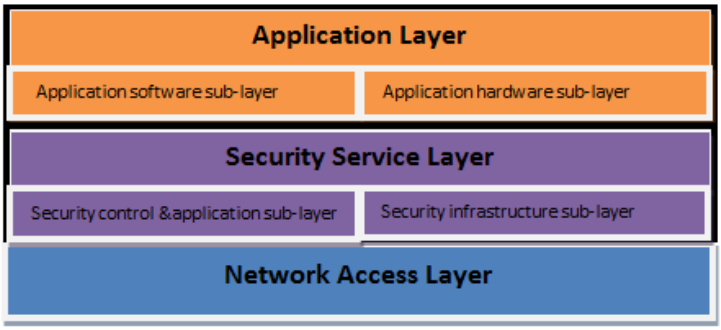

Figure 3. Complete Layer Model for the Electronic Voting System

Table 2. Layers, Sub-layers and Components/Modules of the Electronic Voting Layered Model

S/no	LAYER	SUB-LAYER	COMPONENTS/MODULES
1	Application	Application software	Client side module Tallying server module Voter information database module Election module
		Application physical	Client voting terminal Election server machine Voter database server machine
2	Security service layer	Security control & algorithm	Voter authentication module Device Authentication system Information Encryption module
		Security infrastructure	Token processing system Biometric security system
3	Network Access Layer		

4.1 Application Layer

The Application Layer is logically the first layer in the proposed electronic voting system layered model. The application layer has two sub-layers namely, the application software sub-layer and the application hardware sub-layer. This layer contains modules and components that provide interaction between users, administrators and the electronic voting system.

The application software sub-layer contains the following software modules like;

1. Client side module
2. Tallying module
3. Voter information database module

4. Election module

The application hardware sub-layer contains the following hardware components;

1. Client voting terminal
2. Election server machine
3. Voter database server machine

The Client-side Module

The client side module manages and coordinates the voting terminal with all attached hardware. It is basically the pre-voting management module at the voting terminal machine. The software should be able to provide an interface that can enable users on the voting terminal interact with the voting system. In addition to this, the module loads all the necessary components and services of the voting system according to their sequence of execution. It loads the voter accreditation and authentication services and subsequently loads the election module if voter accreditation and authentication are successful. Any programming language that supports object oriented programming can be used for the design of this module. The components and services that make up the pre-voting stage can be implemented as objects in the client side module.

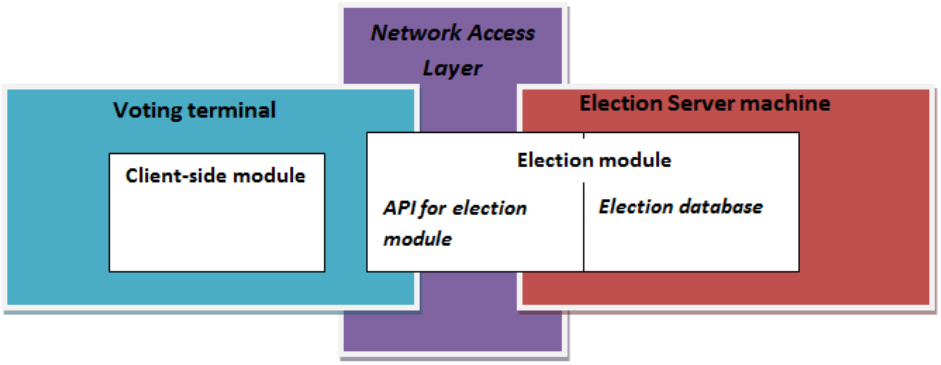

Figure 4. Client-Side Module/Election Module Interaction

The Election Module

The election module is domiciled at the election server machine. This is the main election administration and management module. Once accreditation and authentication are successful, the client-side module connects to the election server machine to download the election module which contains the whole election information like positions for which elections are organized, the candidates and their political parties. The information displayed by the election module is according to the settings and configurations done by an election administrator who activates and deactivates elections. The election module at any given point in time can only display a list of available (active) elections to the prospective voter. The election module would comprise of a suitable application programming interface (API) that interacts with a suitable secure database where election information can be added. The administrator would normally add election information as records of the election entity in the election database from the administrators view or account. Once added, the election information by default would be unavailable until activated by the administrator.

Any suitable programming language system that can integrate well with a database application or query language can be used to implement the election module.

The secure database can be implemented with Structured Query Language (SQL) which is a Fourth Generation Language (4GL). An example of a suitable election entity schema for the election database of the election module will be as follow;

```
Create table Election
(Election_ID                VARCHAR(10),
Election_Definition         VARCHAR(40),
Election_State              VARCHAR(40),
Election_Area               VARCHAR(40),
Primary key (Election_ID))
```

The Tally Module

According to [9], the tallying module receives the ballots from the voting terminal, processes and collates the votes. The tally module should be able to connect to the election module to collect encrypted ballots. It should be able to decrypt the individual ballots and subsequently tally the ballots; and in the case of homomorphic encryption of ballots, decrypt the ballots as a group after tallying the encrypted ballots to produce the result. The tallying module makes the election receipt free, making it impossible to know who has cast what vote. The tallying module design would depend on the ballot encryption module and would have the ballot decryption function embedded on a ballot collation function. The actual vote collation can be done using any suitable mathematical model. The tally module can be implemented using any programming language that support structured or object oriented programming.

The Voter Information Database Module

This module is simply a database system that contains the information of all registered voters. Prospective voters would first supply their biological information on the database during the voter registration window from any designated voter registration points. Before any prospective voter can cast their vote, the system would have to first confirm that they have been registered to participate in the election. This is the voter accreditation phase of voting. The voter information database works with the voter authentication system which connects and relies on the voter information database for the completion of voter authentication.

A sample schema for a simple voter information database would look similar to the following;

```
Create table voter_information
(Voter_ID                          VARCHAR(20) not null,
Surname                            VARCHAR(30) not null,
First Name                         VARCHAR(30) not null,
Middle Name                        VARCHAR(30),
Voter_Picture                      BLOB(1M)    not null,
Sex                                VARCHAR(6)  not null,
Date of Birth                      DATE        not null,
Town                               VARCHAR(30) not null,
Local Government Area of Origin CHAR(30)       not null,
State of Origin                    VARCHAR(15) not null,
Voter_serial no                    INT(20),
Primary Key (Voter_ID),
Foreign Key (Voter_serial no) references Voter_address
Foreign key (Booth_serial no, Polling Booth) references Voting_Area
Check (Sex in ('Male','Female')))

Create table voter_address
(Voter_serial no          INT(20),
Address Line 1            VARCHAR(50) not null,
Address Line 2            VARCHAR(50),
Local Government Area VARCHAR(30) not null,
State of Residence        VARCHAR(20)  not null,
Country of Residence      VARCHAR(20) not null),
Primary key (Voter_serial no))
```



```
Create table Voting_Area
(Booth_serial no          INT(20) not null,
Polling_Booth             VARCHAR(20) not null,
Polling_Unit              VARCHAR(20) not null,
Ward                      VARCHAR(40) not null,
Local Government Area     VARCHAR(30) not null,
State of Participation    VARCHAR(15) not null,
Primary key (Booth_serial no, Polling Booth))
```

Client Voting Terminal

The voting terminal is simply a computer that can allow voters interact with the electronic voting system to cast their votes. It displays all election information on a screen for the prospective voter to complete the voting process. The computer could be a Personal Device Assistant (PDA) with screen size that can allow voters see all display information legibly. The screen may be a touch screen, where voters would access soft buttons and soft keys on the screen or a normal screen where voters would need to use a keypad and pointing device to give inputs to the system. The voting terminal should be selected based on the desired client module; and should run the client-side module optimally. In addition to this, it should also be a network ready computer that can access the selected network access layer infrastructure, and should also support the other hardware peripherals and software modules at the client side.

The voting terminal machine is proposed with optimal Random Access Memory (RAM) size, good graphic memory, a hard disk and a removable storage that stores information. In addition to this, the machine should be disable persons friendly, allowing disabled individuals to make optimal use of the machine. There is no particular specification for the hardware components of the client-side voting terminal. Designers are free to adapt their hardware choice to their system needs.

Figure 5. Diebold Touchscreen Electronic Voting Terminal

Election Server Machine

The election server machine is a powerful computer with high processing power and speed that can run the election module efficiently. The processor should have high speed to support the election module and should be a shared memory chip multiprocessor. The shared memory chip multiprocessor is discussed in [2]. The tally module may be integrated in the election server machine in some implementation, while in some other implementation the tally module may reside in a standalone tally server machine. The hardware configuration of the election server machine should be similar to that of an

equivalent multi-access server machine that can respond to multiple data access requests at a time.

Voter Database Machine

According to [20], a database machine is a computer or special hardware that can store or retrieve data from a database. The database machine proposed in this work is a collection of computers in a database machine network/cluster that will run the voter information database module in distributed database architecture. A distributed database is one in which portions of the database are stored on multiple computers in multiple sites within a network [8]. The essence of having the database in a distributed architecture is to give room for failure recovery. In the event of one of the computers in the database machines network/cluster failing, the database will not be entirely destroyed or lost. This is particularly important especially during the voter registration or voter authentication stages of the voting exercise.

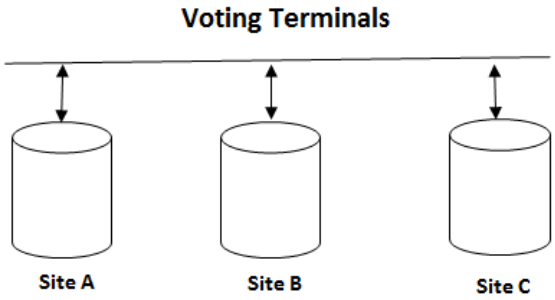

Figure 6. Distributed Database Architecture

In addition to running distributed database architecture, the database system should also implement a redundant array of independent disks (RAID) architecture on each database machine. A RAID 6 system would enable block level striping with double parity for failure recovery. Thus it protects user data when two drives fail at the same time. This is a sharp contrast to a RAID 5 system which adds one parity drive and can only recover user data when one drive fails. The use of RAID 6 architecture ensures an increase in capacity, speed and reliability of the database system.

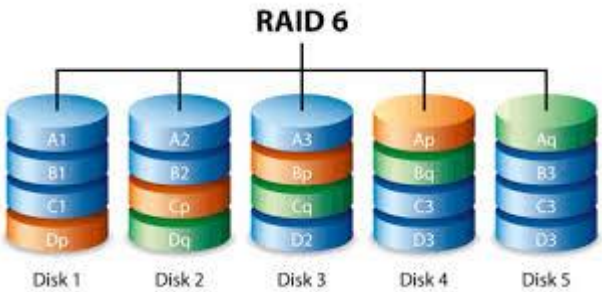

Figure 7. RAID 6 Architecture

5. Result and Conclusion

We have successfully described the application layer of the layered architecture. The application layer will make it easier for voting systems designers to design modular systems. This will allow component level maintenance and design, making it possible to integrate components from different vendors into your voting system implementation. A

well layered architecture also improves the maintainability of the electronic voting system. System designers can choose from a pool of hardware from different vendors when trying to build systems for different clients. Clients can have their preferred system coupled for them according to their needs and budget. Layering the system also helps in ensuring that manufacturers do not hide the source code and design of such systems. Giving sole rights to a single manufacturer could lead to manipulation of the system in order to commit electoral fraud. Cost is another important factor in the proposed model. The proposed model enables a cheaper implementation of the entire system. This greatly affects the cost of maintaining the system downwardly. The second part of this work will focus on the definition and description of the security service layer and the network access layer.

References

[1] D. Lundin, "Component Based Electronic Voting Systems, Towards Trustworthy Elections", Springer-Verlag Berlin, Heidelberg, ISBN:3-642-12979-X 978-3-642-12979-7. Retrieved April 10th 016 from http://research.microsoft.com/en-us/um/redmond/events/wote2007/papers/02.pdf, **(2010)**, pp. 260-273.
[2] D. R. Ham, "Dynamic Scheduling in Multicore Processors", thesis, Faculty of Engineering and Physical Sciences, School of Computer Science, University of Manchester. Retrieved February 10th 2016 from https://www.escholar.manchester.ac.uk/api/datastream?publicationPid=uk-ac, **(2011)**.
[3] G. Stoneburner, C. Hayden and A. Feringa, "Engineering Principles for Information Technology Security (A Baseline for Achieving Security)", Revision A, National Institute of Standards and Technology Special Publication 800-27 Rev A Natl. Inst. Stand. Technol. Spec. Publ. 800-27, **(2004)**.
[4] Intel whitepaper, "Intelligent RAID 6 Theory Overview and Implementation", Intel Corporation. Retrieved March 10th 2016 from www.lamsade.dauphine.fr/~litwin/cours98/Doc-cours-clouds/raid6-intel.pdf, **(2005)**.
[5] J. Shenk, "Layered Security: Why It Works", SANS Institute InfoSec Reading Room. Retrieved March 12th 2016 from https://www.sans.org/reading-room/whitepapers/analyst/layered-security-works-34805, **(2013)**.
[6] M. Ramilli, "Designing a New Electronic Voting System", thesis, University of Bologna Italy. Retrieved May 3rd 2016 from deisnet.deis.unibo.it/nsl/Thesis.pdf, **(2013)**.
[7] M. Rouse, "Layering", Retrieved April 10th 2016 from http://whatis.techtarget.com/definition/layering, **(2005)**.
[8] M. Rouse, "Distributed Database", Retrieved April 12th 2016 from http://searchoracle.techtarget.com/definition/distributed-database, **(2005)**.
[9] N. E. Reginald, "Mobile, Secure E - Voting Architecture for the Nigerian Electoral System" IOSR Journal of Computer Engineering, vol. 17, no. 2, **(2015)**, pp. 27-36.
[10] O. Domy, "Layering or the Modular Approach to Information Systems", Orange Business Services. Retrieved 30th March 2016 from http://www.orange-business.com/en/blogs/connecting-technology/data-centers-virtualisation/layering-or-the-modular-approach-to-information-systems, **(2010)**.
[11] A. M. Oostveen and P. V. Bessdaar, "Users Experiences with E-voting: A Comparative Case Study", Journal of Electronic Governance, vol. 2, no. 4, **(2009)**, pp. 38-46.
[12] J. Orton and K. Weick, "Loosely coupled systems: A Reconceptualization", Academy of Management Review, vol. 15, **(1990)**, pp. 203-223.
[13] G. Z. Qadah and R. Taha, "Voting Systems: Requirements, Design, and Implementation", Computer Standard Interference, vol. 29, no. 3, **(2007)**, pp. 376-386
[14] R. Braden, "RFC 1122, Requirements for Internet Hosts – Communication Layers", Retrieved 5th March 2016 from https://tools.ietf.org/html/rfc1122, **(1989)**.
[15] R. Braden, "RFC 1123, Requirements for Internet Hosts – Application and Support", Retrieved 5th March 2016 from https://tools.ietf.org/html/rfc1123, **(1989)**.
[16] T. Lammle, "CompTIA Network+ Study Guide", Wiley Publishing, Inc., Indianapolis, Indiana, **(2009)**.
[17] Okediran, "A Framework for a Multifaceted Electronic Voting System", International Journal of Applied Science and Technology, vol. 1, no. 4, **(2011)**.
[18] VoteHere Inc., "Network Voting Systems Standards", Public Draft 2, USA **(2002)**.
[19] Wikipedia. Modularity. Retrieved May 10th 2016 from https://en.wikipedia.org/wiki/Modularity
[20] Wikipedia. Database Machine. Retrieved May 12th 2016 from https://en.wikipedia.org/wiki/Database_machine

Authors

Emeka Reginald Nwogu, works with the Michael Okpara University of Agriculture, Umudike Nigeria. He holds a Bachelor of Engineering degree in Electrical and Electronics Engineering (Telecommunication option). He also holds a Master of Science degree in Information Technology. He is a Cisco Certified Network Professional, a Cisco Certified Network Associate, CompTIA Network plus Certified and CompTIA Security plus Certified. He is presently working on his PhD Dissertation at the University of Port-harcourt, Nigeria.

Chinedum Emmanuel Ihedigbo, holds a PhD from Capella University, Minneapolis, MN. He also holds an M. Sc. in Computer Science from Alabama A & M University, Normal, Alabama, with a Bachelors degree from the same university. Presently, he is a Senior Lecturer at the Department of Computer Science, Michael Okpara University of Agriculture, Umudike, Nigeria.