

# Service Oriented Architecture Model for Integration of E-government Systems in Kenya

Joseph Kaibiu Gitau \*, Stephen Mburu

School of Computing and Informatics University of Nairobi, Kenya

\*Corresponding author: [joekaibiu@gmail.com](mailto:joekaibiu@gmail.com)

**Abstract** During the last decade government organisations in Kenya have worked to automate processes and digitize information and services using various systems. These systems have however become diverse due to the various vendors and their use of different data formats, storage types, languages and technologies, thus the issue of heterogeneity and interoperability of systems. This has created a need for an integrated platforms to enhance sharing of information and services between organisations. The E-government platforms in Kenya are growing to a size that requires a framework that ensures an integrated platform for the e-government applications and services provided to citizens, business and other government agencies. Currently most e-government platforms are independent thus result to redundancy of efforts, inconsistency of data and lack of integration, while some platforms are peer-to-peer integrated resulting to tight coupled system, and tedious process of adding of new services into E-government systems. The aim of this research is to use the eCitizen portal in Kenya as a case study, thus understand the challenges that users and technology support staff have with the current e-government systems by use of questionnaires, and use Service Oriented Modelling and Architecture (SOMA) to come up with a Service Oriented Architecture (SOA) Model that can be used meet most of these challenges, and also validate this model using a prototype. The SOA model has shown how we can use SOA to reduced costs, less redundancy of data and effort, shared information and services, interoperability and ultimately better service delivery using a SOA integrated eCitizen platform.

**Keywords:** Service Oriented Architecture, Web Services, e-Citizen portal in Kenya, IPRS in Kenya

**Cite This Article:** Joseph Kaibiu Gitau, and Stephen Mburu, "Service Oriented Architecture Model for Integration of E-government Systems in Kenya." *American Journal of Information Systems*, vol. 4, no. 3 (2016): 59-68. doi: 10.12691/ajis-4-3-1.

## 1. Introduction

The Kenyan government faces an increased pressure to form an effective e-Government that has resulted in vast investments on the ICT infrastructure. The Kenya E-Government Secretariat was set up in 2004 under the Office of the President to be an oversight body to galvanize all ICT projects within government aimed at enhancing service delivery of all the ministries, almost at the same time the Ministry of Information and Communications was set up in 2004, mainly to handle the wider universal access goals to enable the citizens actively participate in a global economy [3]. However, there are issues in the methods, practices, performance and success in managing ICT projects. In the developing countries, projects after projects are being done aimed at bridging the digital gap. The Government of Kenya has implemented electronic systems in various State Departments and other state-owned institutions, including national tax systems, immigration information system, legal information system, the integrated financial management system, education system among other to bridge this digital gap. Most of these systems are to be found in various ministries and departments, and

information and data is manually exchanged between these departments and institutions using fax, e-mail and electronic media. These systems provide partial electronic services to citizens and businesses through Government portals and websites.

This has been brought about by challenges in the integration of this systems due to their heterogeneity and lack of interoperability brought about by the diverse data formats, program languages, data storage methods and vendor's system access to the source code, thus some being closed and other open source. This has in turn resulted to data redundancy, effort duplication, lack of shared information, inconsistent information and long ineffective processes as need to verify details from other departments is basically manual, thus taking days to weeks. The proposed solution to the integration challenges is the use of Service Oriented Architecture (SOA). SOA is an architectural pattern in computer software design in which application components provide services to other components via a communications protocol, typically over a network [8]. The principles of service-orientation are independent of any vendor, product or technology. SOA works around the principles of loose coupling, service abstraction, autonomy, granularity, normalization and re-usability [15].

This paper therefore aims to use the eCitizen portal<sup>1</sup> in Kenya as a case study, thus understand the challenges that users and support staff have with the current e-government systems by use of questionnaires, and SOMA modelling to come up with a SOA Model that can be used to meet most of these challenges. To validate this model using a prototype that was subjected to usability and user satisfaction survey. Finally the paper gives some recommendations focusing on the government institutions in order to improve information and services delivery using the proposed SOA model.

## 2. Literature Review

The evolution in the factors motivating the development of e-Government over time: Initially, the introduction of ICT in government was driven by the need for greater efficiency. Then, the effectiveness of public service provision was included as a goal. And Latter, good governance has been added as an objective in its own right. Most governments aim to empower citizens and businesses, reinforce mobility in the Single Market, enable efficiency and effectiveness, and create the key pre-conditions for e-Government [8].

The Kenyan e-Citizen portal is an online web service that offers both Kenyan citizens and foreign residents various government to Citizen (G2C) services. Some of the G2C services offered include Business Name Registration, Notice of Marriage and Issuance of a marriage certificate, Provisional and Interim Driving License, Driving License Renewal and duplication, Land Rent Clearance Certificate, Passport application and Application for Kenyan Visas. The traditional peer-to-peer e-government integration results in a tightly coupled e-government system which reduces agility, expansibility and interoperability. As a result the systems suffer from several inefficiencies as follows: Firstly, tightly coupled applications. Secondly, in-agility and in-expansibility of the systems; the systems were developed by different developers using different programming languages in different platforms. Thirdly is unreliability in communication between government entities and agencies. To address these challenges, SOA is considered one of the promising technologies that provide interoperability and integration between various services, different software applications, running on a variety of platforms in government and non-government organizations.

### 2.1. System Integration

System integration is defined as putting diverse hardware and/or software components together to work as a system. Integration is a complex task, especially when the applications involved reside on older legacy systems or utilize different hardware or software platforms. From a historical perspective, the earliest computer systems were large stand-alone computers known as mainframes that ran only one computer program at a time. Multiprocessing, the ability to run several programs, each in a distinct partition of the mainframe's memory, was a technical breakthrough that arrived in the 1960s. Since then, businesses have continued to require ever more computing

power and flexibility, and the level of complexity of software solutions has increased significantly.

#### 2.1.1. Levels of System Integration

##### • Data-level integration

The backend data stores of relevant application are integrated, and can be either push or pull based. When using push based, one application makes SQL calls on another application's database tables. This is through database links or stored procedures, and data is pushed into another application's database. However, pull based integration uses triggers and polling. The triggers capture changes to data and write the identifying information to interface tables. It is then possible for adaptors to poll the application's interface tables and retrieve the pertinent data. This pull based integration is used when an application requires passive notification of changes within another application's data. When the application that needs to be integrated does not provide any APIs or client interfaces, you would use data-level integration.

##### • User interface (UI)-level integration

This ties integration logic to user interface code, and can be either scripting or proxy based. When using scripting based, the integration code is embedded into the user interface component events, common with client/server applications such as PowerBuilder. In cases where direct access to the database is not easy or possible, or when the business logic is embedded in the user interface, this is the correct integration method to use. However, user interface level integration is generally used as a last resort. If you add scripting logic to catch events with client/server applications they become very difficult to maintain, as integration levels increase and more changes occur. Furthermore, user interface changes can break integration triggers and logic. This tight coupling creates a permanent link between the maintenance of the user interface and the integration code.

##### • Application-level integration

This is considered the best way forward for application integration, and it uses the integrated application's integration frameworks and APIs. It is good to use, since it is transparent to the integrated application and it preserves the application's data integrity. The application interface allows you to invoke business logic to preserve data integrity.

##### • Method-level integration

This is less frequently used specialization of the application level integration method discussed above. Here, we aggregate common operations on multiple applications into a single application that fronts the integrated applications. It is generally used when each integrated application has a similar set of API or functional methods. The integrated applications must support a Remote Procedure Call (RPC) or distributed component technology. The main disadvantage of this approach is the tight application coupling in front components. They will break when changes are made to the integrated application API, and these problems will propagate down to the other applications that rely on them. This is used when we have distributed component or CORBA technology.

#### 2.1.2. System Integration Architectures

In a well-designed building, the electricians and plumbers usually keep working no matter how many appliances are

<sup>1</sup> <https://www.ecitizen.go.ke/>

switched on. Such a building is also capable of extension without having to tear up the blueprints and start again because of its good architectural design. The same applies to software systems. Software architecture is the backbone of any complex computer system. The architecture encompasses all of the software elements, the relationships between the elements and the user interfaces to those elements. The performance and reliability of a software system are highly dependent upon the software architecture. Well-designed software architecture can be extended with relative ease to accommodate new applications without requiring extensive infrastructure development. Described herein are the most common "Integration" Software Architectures.

- **Point-To-Point Integration Architecture** - The original architecture used to support systems integration is referred to as Point-to-Point. It derives its name from the direct, tightly bound connections that are made between applications and is the simplest of the integration architectures.

- **Hub and Spoke Integration Architecture** - The earliest formal integration technologies worked on the principle that all information coming from the applications had to be processed within a single machine or server called a "hub". Acting as a central point of control, the hub dealt with all message processing including routing, splitting and combining of messages, mapping, and so on.

- **Distributed Integration Architecture** - One solution to the Hub and Spoke scalability issue is to perform message translation, routing, splitting, and combining closer to the source and target systems by using smaller computers known as "agents." Agent computers are connected to just one system and reduce the processing load on that system. Agents take information from the application they are connected to, process it, and send it to any target application(s) interested in receiving that information. The end result is Distributed Architecture. It is also known as **Peer to Peer** architecture.

- **Service Oriented Architecture (SOA)** - Service Oriented Architecture (SOA) is the latest architectural approach, although it's not really very new. SOA is essentially an enhanced version of Distributed Architecture that uses loosely coupled software services to support the requirements of business processes and software users. It goes a step further than the previous architectures by providing an integrated environment which spreads out the workload, breaking down the different "silos" of business functionality and opening their processes to other applications.

### Web Services

The breakthrough for SOA came with the acceptance of Web Services. Web Services was an agreed communication standard between Microsoft and IBM. Based on Web Services, and exhibiting all of the characteristics of the Messaging and BPM solutions previously supplied by the integration vendors, ESB has become the accepted standard for the creation of an organization's Service Oriented Architecture. Without exception all of the integration vendors now provide SOA architecture built on the concept of an Enterprise Service Bus (ESB). When SOA implementation is guided by strategic business goals, chief benefits on an SOA are realized as follows: Alignment of IT with the business and maximal reuse of IT assets. Web services are currently the

most popular technology for implementing a SOA. The basic web services standards are Simple Object Access Protocol (SOAP) for messaging, Web Service Description Language (WSDL) for describing web service interface, and Universal Description, Discovery, and Integration (UDDI) as an optional technology for implementing the service broker. The SOA architecture is illustrated in.

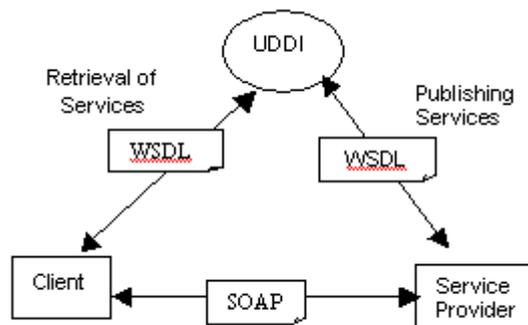


Figure 1. SOA Architecture (Adapted from Shah and Patel [9])

The following are the key aspects of SOA principles [16]:

1. Louse coupling: relationship that minimizes dependency and only requires that services retain an awareness of each other.
2. Service contract: communications agreement, as described in service description.
3. Autonomy: local control over the logic a service encapsulates.
4. Abstraction: hides logic from outside world.
5. Reusability: logic divided into different composable services.
6. Composability: services can be coordinated and assembled to form composite services.
7. Interoperability: open standard-based interfaces and protocols for the plug-and-play architectural components.

## 2.2. SOA Application

### 2.2.1. SOA Perspective of e-Governance in Indian Context

Interoperability has been a critical evaluation criterion for enabling interstate transactions, managing information flow seamlessly and overseeing the backend process for effective delivery of citizen services. In India, e-governance system is still evolving and is not free from challenges as experienced in global terms. There are however many successful ICT initiatives in India oriented towards rural development with a focus to address some specific issues of rural citizens, thus forming "islands". The aim is to provide a portfolio of services to the citizens integrated with e-governance backbone to install a good e-governance system without getting affected during scale up phase. Service orientation is an essential component for Indian e-governance systems since 'desired services' need to be provided to the citizens.

This is possible through a SOA based model which would enable service orientation through citizen demands. Aggregated service demanded are the inputs for the 'service provisioning agencies' in the national network engaged for establishing the orchestrated link to manage the 'service brokering' facility and supply the services.

This provides a scope for the citizens to receive the desired service through SOA based service model. SOA principles draw strength from the benefits of well-practiced architectures in software engineering discipline like client-server, distributed architecture [16]. SOA builds on the strengths of 'application architecture' and 'enterprise architecture' and therefore, has potential to manage e-governance projects. SOA is expected to provide 'universal service identifier' in the system so that desired service can be identified 'on demand' with least transaction time, transaction cost and independent of spatial constraints. It is seen that various service components of SOA can contribute to the Indian e-

governance system in order to provided desired services. The components are 'citizen demand on services', 'service on demand aggregation', 'service-on-supply aggregation', 'service orchestrators' and 'service providers'. A seamless integration of all the services and service provisioning components need to collaborate effectively to focus on citizen centric services. In other words, there are concurrent attempts to provision citizen centric services taken by central and state authorities. Therefore, convergence of services is of prime importance so as to provide commercial approach to the services and establish sustainable and remunerative information service provisioning.

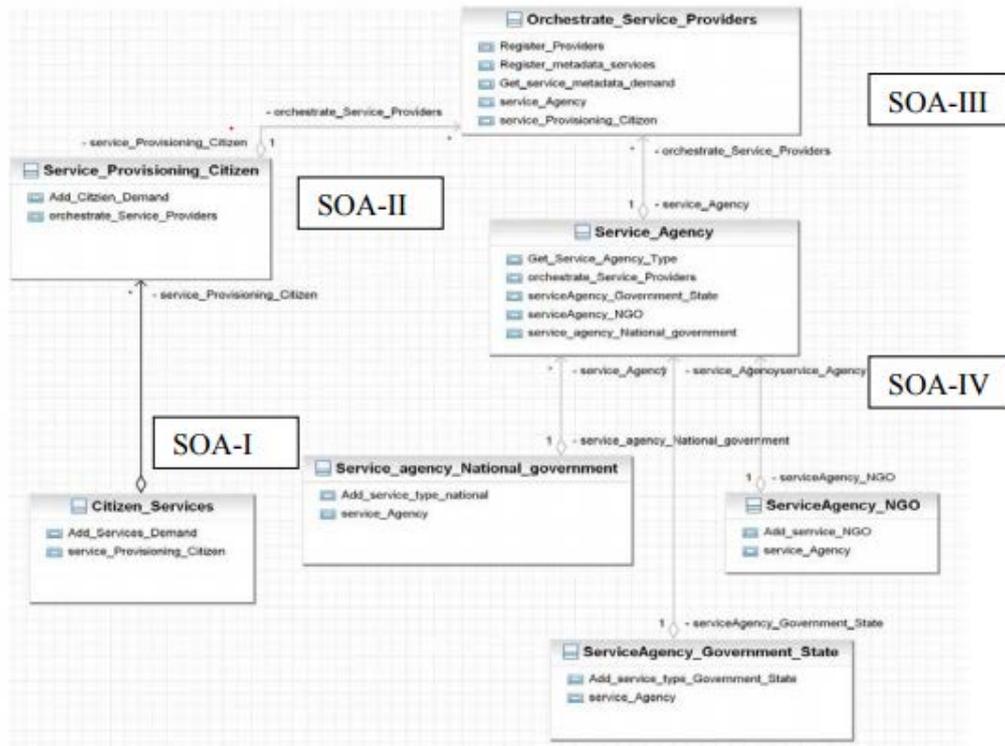


Figure 2. Proposed e-Governance Model (UML based) [5]

SOA architecture based treatment to the Indian e-governance therefore reveal that there is a need to carefully conceptualize and to incorporate all the characteristics of SOA in order to provide citizen centric services as shown in Figure 2 above. It is far more important that countries like India need to carefully articulate services with active collaboration of the citizens in order to deliver good governance systems.

### 2.2.2. M-Shwari Application of SOA

Commercial Bank of Africa (CBA<sup>2</sup>) in partnership with Safaricom<sup>3</sup> Ltd launched a revolutionary product in November 2012 known as M-Shwari. M-Shwari is a suite of banking products that are offered to more than 25 million Safaricom M-PESA customers via their mobile phones. The product has given CBA the platform to offer a much needed service to all market segments through mobile-banking services. These electronic accounts are processed by CBA which allows its customers to operate banking services entirely on a mobile phone, saving them

from visiting a branch. All a customer needs is a mobile handset and be registered on M-PESA. There are no forms or additional documents required to sign up to M-Shwari.

M-Shwari uses SOA architecture as its main integration middleware with other vendors and partners. The SOA architecture provides a seamless real-time exchange of information and up-date of data in the various systems, independent of the applications, data formats, vendors and architectures the various partners and vendors have used.

To successfully launch M-Shwari, CBA required a solution with the ability to seamlessly integrate with its partners, such as Safaricom Limited mobile money platform. Fiorano SOA was chosen as the platform, allowing CBA to expose its Core Banking transactions as web service flows besides ensuring guaranteed message delivery and scalability. Fiorano ESB within Fiorano SOA integrates tightly with the Core banking Integration Framework, thus enabling Real-time update of information between CBA and its vendors, provided a getaway for SMS to be sent out to customer and out-sourcing platform for non-core business added services such as customer care [4].

<sup>2</sup> <http://cbagroup.com/>

<sup>3</sup> <https://www.safaricom.co.ke/>

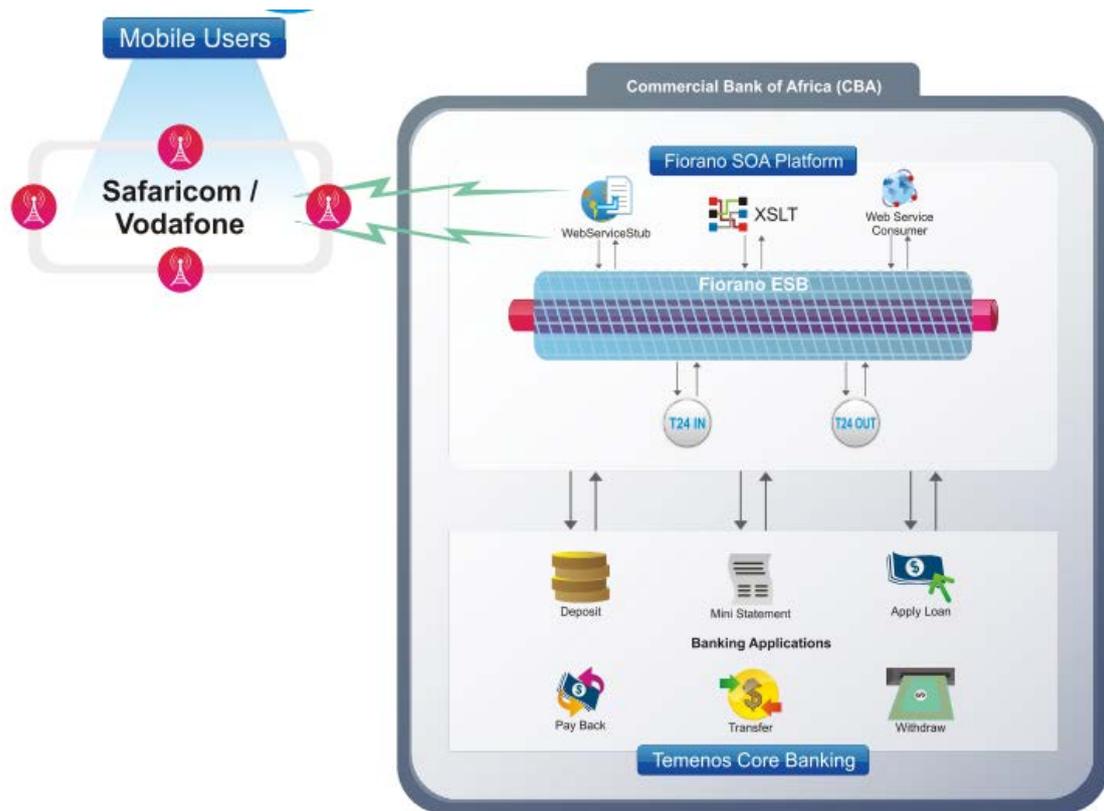


Figure 3. Fiorano SOA deployment on CBA's M-Shwari [4]

2.2.3. An Electronic Government Integrated System Using SOA

The use of SOA model that supports aggregation of information, interaction and personalization of information to

specific user needs can support contextualization and seamless flow of information, thus leading to an improved government information sharing and consequently increasing service delivery to citizens. [11].

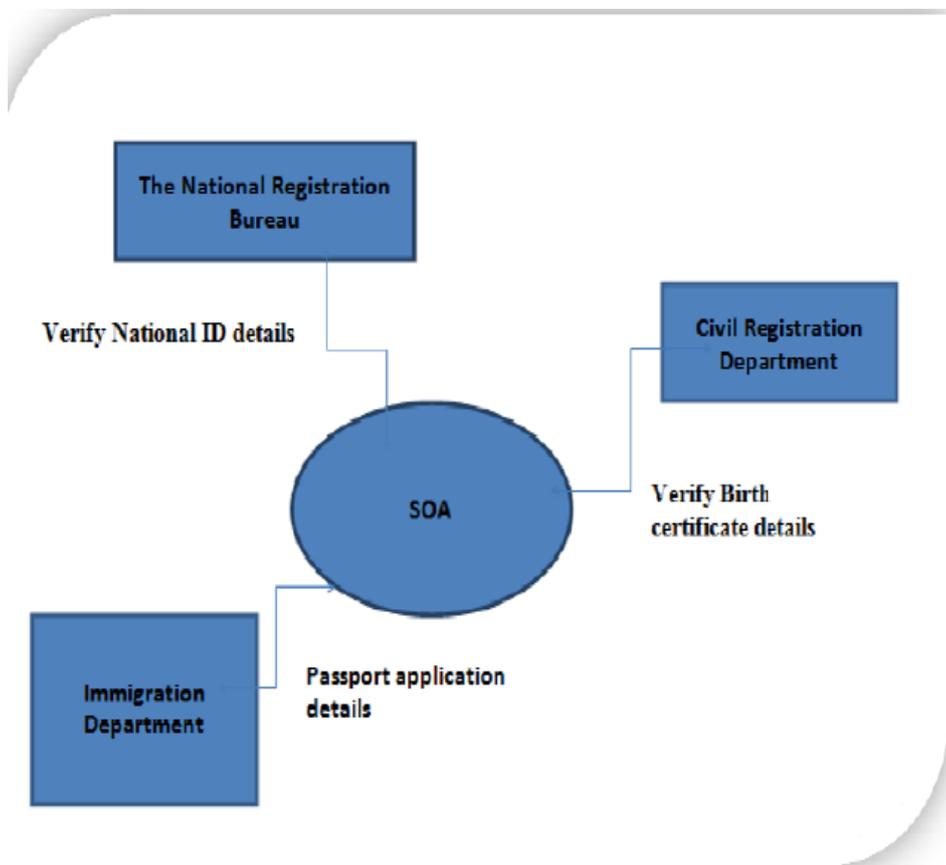


Figure 4. Proposed MoIRP SOA Architecture [11]

The study aimed to investigate the application of SOA as an integration solution to disparate departmental government information systems with the aim of improving service delivery [11].

**Conclusion**

The findings of the study revealed that service delivery can be significantly improved by the adoption of Service Oriented technology. These findings have multiple implications on service delivery and also lowering of costs related to system development.

The study was based on the application of Identification cards and Passports, the took 30 and 14 days respectively, and with use of SOA to integrate the respective

departments reduced this 15 and 7 days respectively by using SOA integration to enhance the Validation stages which were initially done manually using files, and thus improved service delivery significantly.

**3. Methodology**

**3.1. Current eCitizen PORTAL in Kenya**

The current eCitizen portal offers several services from various institutions as summarized in Table 1 below;

**Table 1. Services Provided By The eCitizen Portal In Kenya**

| Services offered in the eCitizen portal in Kenya |   |   |
|--|---|---|
| 1  | Business name Registration              | Registrar of companies                  |
| 2  | Business Name search                    | Registrar of companies                  |
| 3  | Notice of marriage                      | Registrar of companies                  |
| 4  | Issuance of a registrar’s certificate   | Registrar of companies                  |
| 5  | Solemnization of marriage               | Registrar of companies                  |
| 6  | Issuance of a marriage certificate      | Registrar of companies                  |
| 7  | Commissioning of affidavits             | Registrar of companies                  |
| 8  | Special licenses of marriage            | Registrar of companies                  |
| 9  | Provisional Driving License             | National Transport and safety Authority |
| 10   | Driving Test Booking                    | National Transport and safety Authority |
| 11   | Interim Driving License                 | National Transport and safety Authority |
| 12   | Driving License Renewal                 | National Transport and safety Authority |
| 13   | Driving Class Endorsement               | National Transport and safety Authority |
| 14   | Duplicate Driving License               | National Transport and safety Authority |
| 15   | Driving License Information Corrections | National Transport and safety Authority |
| 16   | Official Search (Nairobi Blocks)        | Ministry of Lands                       |
| 17   | Land Rent Clearance Certificate         | Ministry of Lands                       |
| 18   | Application for official copy           | Ministry of Lands                       |
| 19   | Passport application for Adults         | Department of Immigration               |
| 20   | Passport Application for Children       | Department of Immigration               |
| 21   | Application for Kenyan Visa             | Department of Immigration               |

These services are provided in collaboration of various ministries and departments namely Registrar of companies, Registrar of marriages, National Transport and safety Authority, ministry of lands and department of Immigration. A figure of the home page is as below;



**Figure 5.** Current e-Citizen portal

Two studies were done namely

- Questionnaire for users using the eCitizen portal in Kenya
- Questionnaire for IT support staff in the users using the eCitizen portal in Kenya

**3.1.2. Data Collection Methods**

A total of 46 Questionnaires were filled of which 41 were online and 5 were face to face questionnaires. This

was done over a period of 9 weeks and people from different parts of Kenya, and diversity of ages, education, gender, marital and employment status. The Questionnaire had two sections, first section to understand the use and challenges, while the second to measure the usability of the eCitizen portal in Kenya.

**System Usability Scale (SUS)**

SUS is a very easy scale to administer to participants, can be used on small sample sizes with reliable results and its valid, this is, it can effectively differentiate between usable and unusable systems. Interpreting scoring can be complex. The participant’s scores for each question are converted to a new number, added together and then multiplied by 2.5 to convert the original scores of 0-40 to 0-100. Though the scores are 0-100, these are not percentages and should be considered only in terms of their percentile ranking. A SUS score above a 68 would be considered above average and anything below 68 is below average, however the best way to interpret your results involves normalizing the scores to produce a percentile ranking [10].

We used general questions and the System Usability Scale to measure the usability of the current e-Citizen system. The System Usability Scale (SUS) is a reliable, validated and commonly used usability instrument. The instrument provides a global assessment of user satisfaction and allows you to evaluate a wide variety of products and services. The SUS consists of 10 items with five response options ranging from "Strongly agree" to "Strongly disagree". The questions are as follows and were tweaked to suit our case study;

1. I think that I would like to use this system frequently
2. I found the system unnecessarily complex
3. I thought the system was easy to use
4. I think that I would need the support of a technical expert to be able to use this system
5. I found the various functions in this system were well integrated
6. I thought there was too much inconsistency in this system
7. I would imagine that most people would learn to use this system very quickly
8. I found the system very cumbersome to use
9. I felt very confident using the system
10. I needed to learn a lot of things before I could get going with the system

Scoring of SUS is as follows;

- For odd items: subtract one from the user response.
- For even-numbered items: subtract the user responses from 5
- This scales all values from 0 to 4 (with four being the most positive response).
- Add up the converted responses for each user and multiply that total by 2.5. This converts the range of possible values from 0 to 100 instead of from 0 to 40.

The average SUS score is 68. A SUS score above a 68 would be considered above average and anything below 68 is below average. The score is measured as follows;

- 80 or higher is an A - People love your site and will recommend it
- 68 or thereabouts gets you a C. Its OK but needs improvement
- 51 or under gets an F - Make usability your priority now and fix this fast.

From the questionnaire we accumulated the following as the general challenges users had and the necessity to improve the eCitizen portal in Kenya are;

- Slow response time for the website
- Poor marketing of the services - creation of awareness
- Few public services available on the portal - It should be a one stop shop like Huduma Centre<sup>4</sup>.
- Only one access channel - They should also come up with an application for the mobile platform
- Difficulty in navigation - add guidelines and the local language Kiswahili.
- Better customer care - 24/7 and toll free
- Data Inconsistency – Information not updated appropriately in system.

The system usability calculations for the current eCitizen system was as follows;

- Question 1 = 5-1 = 4
- Question 2 = 5-2 = 3
- Question 3 = 4-1 = 3
- Question 4 = 5-2 = 3
- Question 5 = = 4-1 = 3
- Question 6 = 5-2 = 3
- Question 7 = = 4-1 = 3
- Question 8 = 5-2 = 3

- Question 9 = = 4-1 = 3

- Question 10 = 5-2 = 3

**Total = 41\*2.5 = 77.5**

The results of **77.5** means that the website is above average but needs improvements to a standard that people would love it and recommend it to others to use.

### 3.2. Proposed eCitizen portal in Kenya

This paper proposes new integrated e-Citizen portal that gets real-time data requested by users from Integrated Population Registration System (IPRS). IPRS is a central database of all Kenyan citizens and all foreign nationals resident and registered in Kenya.

The use of SOA based integration to integrate the e-Citizen portal and the IPRS system thus data is retrieved directly from the IPRS database through its IPRS Application. This is through the use of SOA application servers and the use of web services to send service requests and expect service responses, through Web Service Definitions Languages (WSDLs).

A service is needed to send a request from e-Citizen system to the IPRS system, and a response is sent back to e-Citizen system to relay the information to the user, and also allow a user to request changes to their information, and provide information and legal documentation used to validate these amendments to the system.

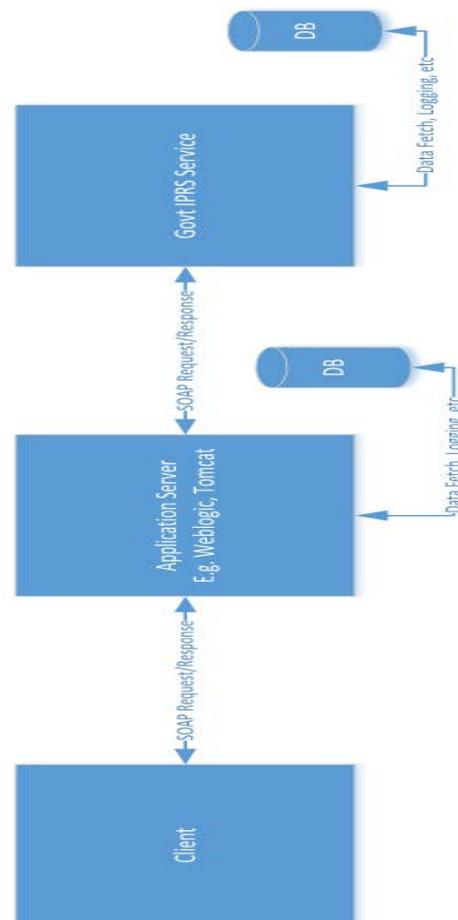


Figure 6. Proposed e-Citizen SOA integration Model

#### 3.2.1. The SOA Model Components

1. **Web Services** - The proposed e-Citizen is deployed as web services that are usable across

<sup>4</sup> <http://www.hudumakenya.go.ke/>

various institutions and individual users. For the prototype used to test the model, we have user **get\_IPRS\_info** as the main service that gets information from the IPRS application server.

2. **WSDL** - The Web service definition language ensures the web services are described in detail in XML and the requests and responses are in a similar format. It creates and validation requests and responses to and from an application, as it defines the data required in the request and the expected response.
3. **BPEL** - The concept uses business process execution language to link the web services together, and define how they interact with each other. The BPEL in our prototype is contained in the Enterprise service bus which we are using in our prototype as the Oracle service bus.
4. **ESB** - The enterprise service bus is used to manage the messages between the many web services and various applications calling these web services. Its main functions will be to monitor and route the messages, Resolve contention, Version web services, Event and Exception handling, and security and validation. For our prototype we are using the Oracle Service bus.
5. **SOAP** - SOAP is an XML-based messaging protocol that defines a set of rules for structuring messages that can be used for simple one-way messaging but is particularly useful for performing RPC-style (Remote Procedure Call) request-response dialogues. This is used as a security measure for intruders who want to intercept data.

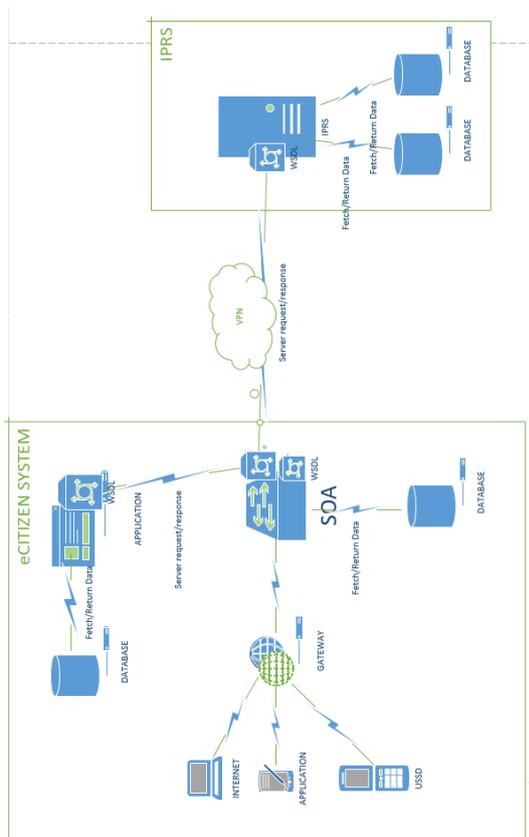


Figure 7. Proposed SOA Model

## 4. Results

### 4.1. Post Prototype Questionnaire

The System Usability Scale was used again on a random sample of 12 users on the proposed prototype and the scale was calculated as below

- Question 1 = 5-1 = 4
  - Question 2 = 5-2 = 3
  - Question 3 = 4-1 = 3
  - Question 4 = 5-1 = 4
  - Question 5 = 5-1 = 4
  - Question 6 = 5-2 = 3
  - Question 7 = 5-1 = 4
  - Question 8 = 5-2 = 3
  - Question 9 = 4-1 = 3
  - Question 10 = 5-1 = 4
- Total = 35\*2.5 = 87.5**

#### 4.1.1. Service Authentication Testing

This involved testing whether prototype allows access to unauthorized users. Ensures access only given when appropriate certificates or/and credentials that are provided. See the outcome in Table 2 below

Table 2. Service Authentication Testing Outcome

| SAMPLE   | Type of authentication    | Response      |
|----------|---------------------------|---------------|
| Sample 1 | Right user Right Password | Logged In     |
| Sample 2 | Right user Wrong Password | Failed Log in |
| Sample 3 | Wrong user Right Password | Failed Log in |

#### 4.1.2. Service Validation Testing

This involves testing whether the prototype is able to validate the data provided in the web request before sending to the respective application server. See the outcome in Table 3 below

Table 3. Service Validation Testing Outcome

| SAMPLE   | Type of Validation | Response |
|----------|--------------------|----------|
| Sample 1 | Valid ID           | Response |
| Sample 2 | Invalid ID         | Error    |

#### 4.1.3. Communication Channel Testing

The communication channels are specified in the prototypes and each service specify the addresses it remotely connects to, this is, request coming from and is sent back to. See the outcome in Table 4 below

Table 4. Communication Channel Testing Outcome

| SAMPLE   | REQUEST    | CHANNEL | RESPONSE |
|----------|------------|---------|----------|
| Sample 1 | Complete   | Website | Response |
| Sample 2 | Incomplete | Website | Error    |
| Sample 3 | Complete   | Code    | Response |
| Sample 4 | Incomplete | Code    | Error    |

#### 4.1.4. Service Request Testing

Service Request Testing - This involved testing whether the system gets a service response for each service request. See the outcome in Table 5 below

Table 5. Service Request Testing Outcome

| SAMPLE   | Request  | Response      | Comment |
|----------|----------|---------------|---------|
| Sample 1 | right ID | Info received | Worked  |
| Sample 1 | wrong ID | Info received | Worked  |

#### 4.1.5. Test Response Time Testing

Test Response Time Testing - The time taken for the application to respond to requests. This is time between when web request is sent and web response is gotten. See the outcome in Table 6 below

Table 6. Test Response Time Outcome

| SAMPLE   | TIME TAKEN     |
|----------|----------------|
| Sample 1 | 1sec 259 msecs |
| Sample 2 | 1sec 298 msecs |

### 5. Conclusion

The SOA model proposed would make use of minimal additional resources and software, and would also make use of the current systems for its adoption. From the layout and implementation the model if implemented will sort some current challenges noted in the questionnaires like the

- Easier addition of services thus more services provided.
- Provision for more channels, like mobile applications and user menus.
- Consistency and Integrity of data, and
- Minimizes manual processes like dumping and uploading of files.

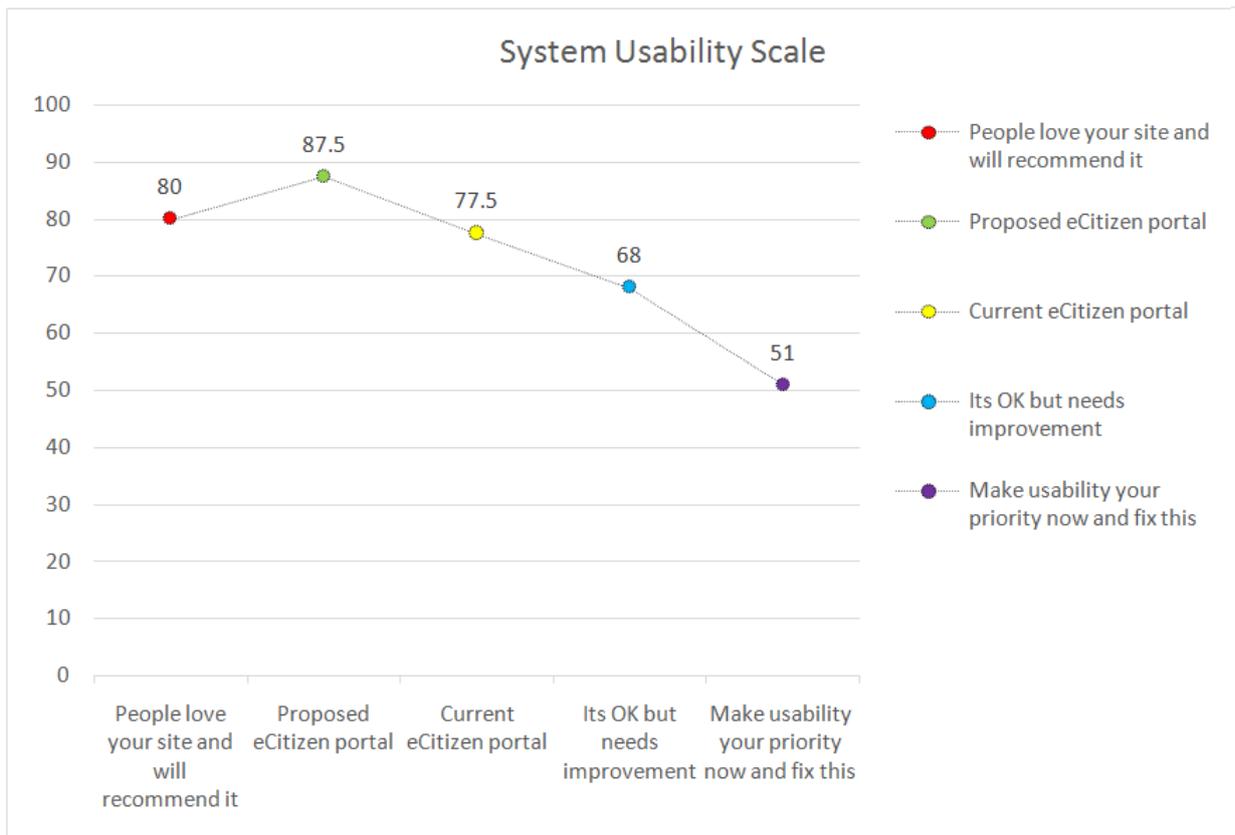


Figure 8. System Usability Scale comparison

#### 5.1. Potential Issues by the Introduction of the SOA Model.

The introduction of the SOA model to the eCitizen portal in Kenya would come with some potential issues that would include, but not limited to;

- **Security** - The security of the SOA model is a multiplexed matrix that includes the security policies, software and hardware of the connected parties, including the service provider, service user, internet providers and other infrastructure.
- **Infrastructure** – The SOA Model is dependent of the internet and its reliability and speeds, the electricity and national grid, and other structural dependencies.
- **Expertise and Skill** – The SOA Model is also dependent of available expertise and skill

available in-house or for consultancy, and their availability for the design and implementation of the model, and even most important the maintenance in terms of monitoring, upgrades, structuring and troubleshooting.

- **Legal issue** – The SOA Model works around sharing of data and information between institutions both governmental and private, each of which have their own laws and policies of sharing such data and information.

#### 5.2. Recommendations

Based on the study findings the following are the recommendations

- Government institutions should do more research and study on better integration methods like Service Oriented Architecture.

- Government institutions should adopt the use of Service Oriented Architecture for effective and efficient integration of various services.
- Government institutions should collaborate with the private sector for better services provision and delivery.
- Concerns on the use of Service Oriented Architecture should be considered in the development of the ICT strategy. The concerns include but not limited to security, privacy, legal issues, infrastructure, affordability of services and service delivery.

## Acknowledgement

The users and support staff from various institutions that participated in the survey used in this paper.

## References

- [1] David, S, Monica, K (2012) INTERNET GOVERNANCE IN KENYA – AN ASSESSMENT for the Internet Society.
- [2] International Telecommunication Union (ITU) (2014) *ICT Facts and figures 2014*.
- [3] ICT AUTHORITY KENYA (2014) THE KENYA NATIONAL ICT MASTERPLAN 2013/14 – 2017/18.
- [4] Fiorano (2015) M-Shwari logs 40,000 customers per day crossing Ksh1 billion in transactions a month with core banking integration from Fiorano (<http://www.fiorano.com/customers/casestudies/commercial-bank-of-africa.php>).
- [5] Harekrishna, M.(2009 ).Understanding SOA perspective of e-Governance in Indian Context: Case base study.
- [6] Government of Kenya *Integrated data system to make e-government a reality* (<http://www.president.go.ke/2015/03/11/integrated-data-system-to-make-e-government-a-reality/>).
- [7] Muigai, Alfred, N (2012) *Challenges facing e-government projects in Kenya*.
- [8] P. Wauters, K. Declercq, S. van der Peijl, P. Davies (2012) Study on cloud and service Oriented architectures for e-government for Deloitte.
- [9] Shah, D., & Patel, D. (2009). *Architecture framework proposal for dynamic and ubiquitous security in global SOA. International Journal of Computer Science and Applications*.
- [10] Sauro, J. (2011). Measuring Usability with the System Usability Scale (SUS).
- [11] Violet L. (2014) an electronic government integrated system using service oriented architecture (SOA).
- [12] Kenya eCitizen Portal <https://www.ecitizen.go.ke/ecitizen-services.html>
- [13] Kenya immigration department [http://www.identity.go.ke/index.php?option=com\\_content&view=article&id=90&Itemid=115](http://www.identity.go.ke/index.php?option=com_content&view=article&id=90&Itemid=115).
- [14] Liu, M., (2009) “NET BPEL Microsoft SOA Web Services”. <http://www.packtpub.com/article/soaservice-oriented-architecture>.
- [15] Oracle SOA – Service Oriented Architecture <http://www.oracle.com/us/products/middleware/soa/overview/index.html>.
- [16] SOA Life cycle <http://searchsoa.techtarget.com/tip/Understanding-the-SOA-lifecycle>.
- [17] SOA-Glossary, *Cambridge Technology Enterprises*”, available from <http://www.ctepl.com/soaterms.shtml>.
- [18] Zimmermann, O., *et al*, “Analysis and design techniques for Service-Oriented Development and Integration”, available from <http://www.perspectivesonwebservices.de/download/INF05-ServiceModelingv11.pdf>.