TECHNOLOGY TRENDS FOR SPATIAL DATA INFRASTRUCTURE IN AFRICA

Collins Mwange[1], Prof. Galcano C. Mulaku[2] , Dr. Ing. David N. Siriba[3]

[1]University of Nairobi, Kenya, cmwange@gmail.com

[2]University of Nairobi, Kenya, gmulaku@yahoo.com

[3]University of Nairobi, Kenya, dnsiriba@uonbi.ac.ke

Abstract

Over the past few years, several technology trends notably Cloud Computing, Volunteered Geographic Information (VGI), Free and Open Source Software (FOSS), Internet of Things (IoT), Big Data and Linked Data have emerged. Such technologies have great potential for supporting wider adoption of Spatial Data Infrastructures (SDIs). Coupled with mature industry standards such as the Open Geospatial Consortium (OGC) Web Map Service (WMS) and Web Feature Service (WFS), there could be no better time that African countries can to hasten development of their national SDIs. This study reviews contribution of the new trends to SDI development in Africa, with a view to answering the research question: *how can technology trends promote development of SDI in Africa*? A simple geospatial application based on Google Cloud Services (GCS) was developed. Specifically, the application was based on Google Container Engine (GKE), an Infrastructure as a Service (IaaS) cloud. Data was sourced from the 2015 Kenya Certificate of Primary Education (KCPE), Ministry of Education, Science and Technology (MOEST) School mapping data of 2007 and shapefiles of Kenya's key administrative boundaries from the Independent Electoral and Boundaries Commission (IEBC). Using several FOSS software in the cloud environment, several operations and analyses typically common in SDIs were carried out. The study shows that cloud computing can increase uptake of SDIs, through highly scalable web services running in the cloud. Additionally, the study shows that FOSS software, which was used extensively, has great potential for SDIs particularly in developing countries where resources are limited.

Keywords: Spatial Data Infrastructure, Africa, GDI, SDI

1. INTRODUCTION

1.1. Background

It is often argued that technical components are easier to deal with than the non-technical ones when developing a Spatial Data Infrastructure (SDI) (Budhathoki and Nedović-Budić, 2007). Technology components, which are part of technical components, include standards, access networks, clearinghouse portals and policies.

In spite of being easier to deal with, a significant amount of resources are often invested on technical components. More significantly, technical components are the most visible in SDI, facilitating effective communication with decision makers who are keen to ascertain the tangible benefits of projects through analysis techniques such as cost-benefit analysis (CBA).

There are several reasons why consideration of technology components is important in SDI. First, distributed network access (through various applications and geoportals) is what makes spatial data more readily available to end users. Secondly, advances in Information and Communication Technologies (ICT) is a major driver shaping and influencing SDIs. Thus ICT offers immense opportunities to improve SDIs. Finally, the identification of trends that impact SDI can help in development of policies to regulate and deal with the issues presented (GeoConnections, 2013).

The development of applications allowing ordinary citizens to participate and contribute to web mapping activities (Newman et al., 2010) is an important emerging trend. Notable examples of applications following this framework include the OpenStreetMap (OSM), Wikimapia and Google Earth (Goodchild et al., 2007). Some of these frameworks can be considered as forms of VGI. Such applications fit within the framework of web 2.0, which provides bi-directional collaboration in which users are able to interact with and provide information to central servers and sites (Goodchild et al., 2007). Web 2.0 describes the shift from a web of documents and users as passive consumers to a broader platform for communication, collaboration and business transactions. In turn, this strengthens the role of citizens as SDI stakeholders (Díaz et al., 2012). The bottom-up approach to SDI development fits perfectly well within this framework, since it recognises the contribution of ordinary citizens.

Another emerging trend influencing SDIs is cloud computing. Simply put, a cloud is a utility based computing model that provides a service allowing virtualised or container-based resources to be rapidly and efficiently scaled on demand (Ludwig, 2012). Cloud computing enables ubiquitous, convenient, elastic, scalable, on-demand access to shared pool of resources, such as computing, networks, servers, storage, applications and services. Within the cloud ecosystems, there is a shift from virtualised to container-based cloud services.

Geoprocessing services, which have tended to be slow within contemporary SDIs, can be implemented much more efficiently using cloud computing to realise highly available and scalable services. Additionally, cloud computing can help Geospatial professionals focus on their core domain, with routine IT tasks being managed by the cloud provider (Díaz et al., 2012).

The preceding paragraphs illustrate a few of the technology trends with potential to influence SDI. The objective is to establish whether such technologies have potential to promote wider adoption of SDIs in Africa.

1.2.   Statement of the Problem

Whilst many nations especially those in developing countries are still struggling to develop their NSDIs, technology that supports SDI is advancing at breath-taking pace. For instance, since the United States NSDI was initiated, technologies for data sharing have advanced in leaps and bounds, rendering obsolete the first generation SDIs (Maguire and Longley, 2004). Some researchers have questioned the dominant SDI models, notably the top down model, which renders SDIs less capable of evolving with emerging technologies (Díaz et al., 2012).

There are many challenges affecting current SDIs, serving as a stark reminder that SDI is complex and careful approach is needed in its design. Some of these challenges are: diverse user requirements which are often demanding and unknown, the need for SDI to support new and challenging functionality, user tasks which are unfamiliar to application developers of the SDI (Newman et al., 2010), lack of SDI interconnection and scalability caused by inadequate connectivity between existing SDIs, and due to reliance on ageing technologies (Díaz et al., 2012). The emerging technologies have potential to alleviate some of these challenges.

An SDI is useful if it has a significant and growing installed base, in terms of actively used components and users. This helps the SDI to gain both acceptance and a momentum for self-reinforcing growth (*bootstrapping*) and the ability to adapt to new and improving technologies (*adaptability*) (Díaz et al., 2012). The SDI must integrate usability and user feedback mechanisms into application design and development. Emerging technology components offer great potential to improve the current SDIs. This paper highlights some of new technology trends, as well as technical choices in architecting SDI systems. Developing countries can use the new technologies to speed up implement their SDI.

Unfortunately for Africa, many countries do not have operational (i.e. with strong inter-agency collaboration with an online geoportal) NSDIs yet, making it a difficult study environment. It is for this reason that a simple geospatial application is used to demonstrate the concepts. This study attempts to bridge the gap by reviewing the contribution of new technology trends to SDI development in Africa, with a view to answering the research question: *To how can technology trends promote development of SDI in Africa*?

1.3.   Research Objectives

The objective of the paper is to review the technology trends (e.g. linked data, cloud computing, Representational State Transfer (REST), VGI, web 2.0) and how they can be embraced to promote wider adoption of NSDIs in Africa. Where possible, the contribution and challenges of these technology trends is reviewed.

1.4.   Organization of the Paper

The rest of this paper, which is structured in four chapters including this introductory chapter, is organized as follows. Chapter two reviews literature related to IT trends in SDI, focusing more specifically on trends that African countries can use to promote wider adoption of their NSDIs. Chapter three discusses the methodology adopted in the study. Finally, the results, discussions and conclusions from the study are presented in Chapter four.

2. LITERATURE REVIEW

2.1. Introduction

This section reviews literature related to IT trends that have potential to contribute and positively influence SDIs.

2.2. The IT Building Blocks of an SDI

2.2.1. *Service Oriented Architecture (SOA)*

Service Oriented Architecture (SOA) is a design pattern that focuses on creating generic services that can be reused across many distributed applications. SOA is based on the assumption that all components can be built as services, where SOA facilitates service registration, discovery, and binding to form new functional applications (Yang et al., 2010).

Two notable implementations of SOA are Simple Object Access Protocol (SOAP) and REST. REST is a set of architectural principles that transmit data over Hypertext Transfer Protocol (HTTP) using request parameters, treating data as a resource which in turn is represented uniquely by a Uniform Resource Identifier (URI) (Steiniger and Hunter, 2012). SOAP, On the other hand, is a specification used for exchanging structured information in eXtensible Mark-up Language (XML) format using the HTTP POST request. REST based implementations tend to be less complex than SOAP, implying that SDI can derive more advantages through REST.

2.2.2. *OGC Web Services*

Web services are self-contained and self-describing web applications that can be invoked over the web using messages encoded in XML and transmitted over HTTP (Maguire and Longley, 2004). Spearheaded by the Open Geospatial Consortium (OGC), the OGC web services can be classified into (Li et al., 2013):

- *data visualization*, such as Web Map Service (WMS),
- *data services,* such as Web Feature Service (WFS) and Web Coverage Service (WCS),
- *processing services,* such as Web Processing Service (WPS),
- *data presentation*, such as Styled Layer Descriptor (SLD),
- *querying and transport*, such as Geography Mark-up Language (GML),
- *publication and discovery,* such as Catalogue Service for Web (CSW),
- *data storage*, such as Web Map Context (WMC)

2.2.3. Web Map *Service* (WMS)

Web Map Service (WMS) is an OGC standard for data visualization that enables visual overlay of complex and distributed geographic information over the internet. WMS returns a geo-referenced rasterized map, which is a two-dimensional visualization of styled features and can be delivered using common formats such as Portable Network Graphics (PNG), Joint Photographic Experts Group File Interchange format (JPEG) or Geographic Tagged Image File Format (GeoTiff) (Steiniger and Hunter, 2012). In an SDI, WMS may help in discovery and visualisation of spatial data held in catalogue services (GeoConnections, 2013). This is because WMS returns a map rendered from the data (Li et al., 2013) which is ready for display. The key strength of WMS is that rendering takes place on the server and the output of the request is ready for display on clients in an appropriate format.

2.2.4. Web Feature *Service* (WFS)

WFS is a web service designed for standardized request and return of vector data without any hints as to how the data should be rendered. This in turn facilitates sharing and manipulation of geospatial data. WFS offers direct access to geographic information at the feature and property level (GeoConnections, 2013). Thus, WFS can be used for selecting, inserting, updating and deleting features, as well as geographic and attribute filtering of geospatial features. It differs from a WMS in that it provides access to discrete objects, i.e. vector features, and not just a map ready for rendering (Steiniger and Hunter, 2012).

2.2.5. Web *Coverage* Service (WCS)

Web Coverage Service (WCS) does the same for raster data as WFS does for vectors: it returns the raw raster data instead of the portrayal data. WCS provides available data together with detailed descriptions and defines a rich syntax for requests against the data, and returns the data with its original semantics that may be interpreted, extrapolated, and so on (GeoConnections, 2013). WCS is similar to WFS in that it provides direct access to geographic features, but unlike the WFS that returns discrete geospatial features the WCS can return either whole coverages, a set of features, or a grid coverage. A grid coverage can be an aerial photograph, satellite imagery, or elevation data that typically represent space-varying phenomena (Steiniger and Hunter, 2012).

2.2.6. Web *Processing* Service (WPS)

WPS is a paradigm shift from *data* providing services, such as WCS and WFS, to *service* providing services (Li et al., 2013). WMS uses an approach that publishes and executes geoprocesses over the internet (Brauner et al., 2009). WPS can facilitate sharing, discovery, re-use and dynamic binding of geospatial processes, using a standardized interface to publish and perform geospatial processes over the web (Baranski, 2008). WPS can also facilitate service chaining for tackling complex processes through a sequence of sub-processes. Examples of WPS systems include the Java-based 52°North (www.52north.org/wps) and the Python-based PyWPS (http://www.pywps.org/) (Brauner et al., 2009). Although written in different languages, both are implementations of the WPS service standard from OGC, enabling standard deployment of geo-processes on the web.

2.2.7. Other Services

Whilst data discovery, visualization and access services are fundamental to SDIs, other application services are also needed. These services include application services, catalogue or registry services, portrayal services, and processing services (GeoConnections, 2013).

2.2.8. Geoportals

Portal is a word derived from the Latin word porta, which means a doorway, gate or entrance. Therefore, webportals are web sites that act as a gateway to a collection of information resources, including other data sets, services, cookbooks, news, tutorials, tools and an organized collection of links to many other websites usually through catalogues (Maguire and Longley, 2004). Similarly, geoportals (or geospatial portals) are gateways to geospatial information.

Commonly known as clearinghouse portals (Executive Office of the President, 1994), geoportals are websites where geospatial resources (geospatial data, web services and other geospatial resources) can be discovered (Maguire and Longley, 2004), making it easier for users to find, access and use that information. Geoportals are an important and highly visible component of SDI, serving as the "face" of the SDI.

A Geoportal, illustrated in Figure 1, is typically used as a single window into the SDI (GeoConnections, 2013), where applications address a span of user needs, ranging from generic end-user requirements to the needs of suppliers who contribute data and services. A rich set of applications based on core SDI components, interfaces and services, built by developers and suppliers, ultimately deliver most of the anticipated benefits of the SDI.

From a technology point of view, the ideal SDI exhibits a distributed architecture, with cooperating organizations exposing their data and applications to the SDI. This architecture enables independent systems to communicate and collaborate with one another (Tumba and Ahmad, 2014). Web services based on open standards provide the basis for interaction amongst various applications, allowing users and even other applications to contribute, search, access, exchange, and use geospatial data. Together with the industry, these organizations have collaborated to define various important standards.

Figure 1: Schematic Illustration of a clearinghouse portal
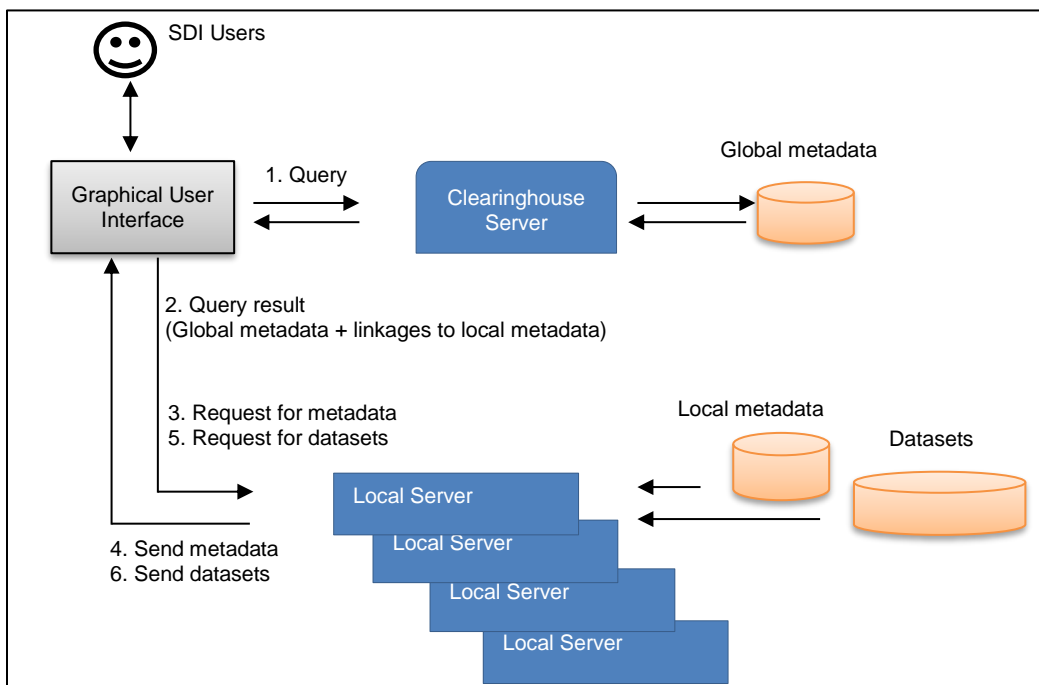
2.3.   The Software Architecture of an SDI

A software architecture can be defined as the structure of a system, usually formed by elements (or components), their properties and the relationships among them and possibly with their environment (Béjar et al., 2009). Most contemporary SDIs are based on the Reference Model of Open Distributed Processing (RM-ODP) standard as the conceptual framework (Henricksen, 2007). RM-ODP, jointly developed by ISO and International Electro-technical Commission (IEC) for architecting open and distributed processing systems, is an international standard that provides a conceptual framework for building complex systems in an incremental manner. The RM-ODP defines five viewpoints on the system and its environment. These are (Henricksen, 2007):

- *enterprise*: focuses on the purpose, scope and policies of the system,
- *information*: focuses on the semantics of the information and processing performed,
- *computational*: functional breakdown of the system into interacting objects,
- *engineering*: mechanisms supporting distributed interaction between objects in the system,
- *technology*: focuses on technology choices in the system.

The software architecture of an SDI is an important consideration to the extent that it facilitates its design and understanding. The architecture is usually documented by a set of views, viewtypes and styles (Béjar et al., 2009). Whereas views are representations of elements and their relationships, viewtypes are allowed element types and relationship types. A style is a specialisation of a viewtype that may, for example, specify that only certain elements and relationships from a viewtype are allowed. RM-ODP defines views of a distributed system, and provides a solid architectural framework upon which SDIs can be built (GeoConnections, 2013).

2.4. IT Trends Influencing SDIs

Currently, there are several technology trends, such as Internet of Things, Linked Data, the Semantic Web and Cloud Computing, all of which have potential to offer new approaches to enhance wider adoption of SDIs. These trends complement the OGC's *Web Services* standards with new ways that can enhance the SDIs. This section reviews some of these trends.

2.4.1. Internet of Things

Internet of Things (IoT) usually involves sensors embedded in appliances and electronic devices, which can be connected to each other through Bluetooth, mobile phone or Wireless Fidelity (WIFI) networks (Wainainah, 2015). Real-time and high-spatial-resolution data can be provided by various geosensors, ranging from weather stations, marine sensors, unmanned aerial vehicles and satellites. To integrate such geosensors and their data with SDI, the Sensor Web Enablement (SWE) technology has been developed by the OGC (Díaz et al., 2012).

IoT facilitates improved means for collecting and accessing near real-time information, and therefore can significantly contribute to SDIs through availability of near real-time spatial data.

2.4.2. Linked Data

Linked Data, which involves a simple representation of the data using a Resource Description Framework (RDF), refers to a set of practices for publishing and connecting structured data on the semantic web. This facilitates linking of data to other pieces of data, thus contextualising and adding value to the information that already exists (Carpenter and Snell, 2013). While the World Wide Web (WWW) is a web of interlinked documents using HTML, the semantic web is a web of interlinked data where RDF is the language for representing the linked data (Brink et al., 2014).

Technically, linked data is machine readable data that is linked to external data sets using the RDF format. The goal of linked data is to create a single global data space on the web.

Linked data is underpinned by four principles:

- utilization of URI to name resources or objects on the web,
- use of HTTP URI's so that people can look up those names,
- URI should return useful information when looked up by RDF and Simple Protocol and RDF Query Language (SPARQL),
- links to URI's allowing discovery of more things

GML, which is the publication of spatial data according to predefined data specifications, is a common data sharing format in SOA-based SDIs. Linked data differs from GML in that it facilitates an open publication environment in which additional information and data from other sources can be easily linked. It is possible to transform existing data from GML to RDF using eXtensible Stylesheet Language (XSLT) transformation, facilitating widespread publication of the data as linked data within existing SDIs (Brink et al., 2014).

According to Díaz *et al* (Díaz et al., 2012), Linked Data offers several benefits to SDIs: simplified integration of heterogeneous data through increasingly shared vocabularies (thus increased availability of information resources), improved means for encoding, describing and interlinking data (hence improved access to data through links and crawling mechanisms), and homogeneous model for data and metadata (thus improved descriptions of data resources and their quality).

2.4.3. Big Data

The term Big Data became accepted in 2010, although it may have existed prior to this albeit with different terminology. Generally, it refers to the high-volume, high-velocity, high-variety, high-veracity, high-value information that require new forms of processing, insight discovery and process optimization (Tsinaraki and Schade, 2016).

Big data may contribute to SDIs by facilitating the storing and processing of geospatial data through cloud computing and analytics, and as a new source of innovation by leading to new geospatial-specific solutions, new bodies of knowledge, new scientific communities, and new specialized conferences (GeoConnections, 2016). Big Data often comes from the cloud, and SDIs may contribute to Big Data in the following ways:-

- spatial and temporal references leads to the creation of new analytical possibilities and the discovery of new facts, which can contribute significantly Big Data analytics
- a large number of Big Data sources include a spatial reference (e.g., latitude-longitude), which can serve as a an integrator or an aggregator with other information sources
- geovisualization, which provides more insight to decision-makers and facilitate the analysis of geographically distributed phenomena, using 2D maps or 3D perspective views

2.4.4. Open Data

Open Data, which maybe geospatial, government, scientific or historical data, is the initiative and idea of free accessibility and availability of data for everyone. Generally, opening data makes it accessible for use to other purposes than it was intended for. Data is open if anyone is free to use, reuse, and even redistribute it.

The main difference between SDI and Open Data is that agreements on commonly used standards and technologies are widely missing in Open Data, and additionally Open Data focuses on content rather than on infrastructure and interoperability. However, both initiatives overlap and complement each other.

2.4.5. Cloud Computing

Cloud computing describes an approach to which applications, services and datasets are no longer located on local computers, but distributed over remote facilities (Díaz et al., 2012). It is a combination and evolution of existing technologies such as grid computing, utility computing and virtualisation (Ludwig, 2012). A *grid* is a network of spatially distributed computation or data resources, which is accessible via open and standardized interfaces (Baranski, 2008).

Virtualisation is an abstraction process that creates a virtual, rather than actual, instance of something else such as an operating system. Cloud computing is a true facilitator for Big Data projects (GeoConnections, 2016).

Clouds can be deployed as private (operated solely for one organization), public (hosted elsewhere in a shared manner and made available to the general public), or hybrid (a combination of private and public clouds, bound together by open or proprietary technology), providing a means to host and serve significant volumes of data without the accompanying capital investment (Carpenter and Snell, 2013). This is ideal for SDIs which may start operations without investment in hardware and associated costs.

Cloud computing provides flexible, location-independent access to computation, software, data and storage resources that are quickly allocated or released in response to demand (GeoConnections, 2013). Clouds can be deployed at three service levels as illustrated in Figure 2. In decreasing order of control and increasing order of security, these levels are (Ludwig, 2012):

- *Infrastructure as a Service, IaaS*, offers complete access to basic computational facilities, the hardware and technology, and virtual machines. Users havefull control over operating systems, storage, and deployed applications. Example of IaaS clouds include the **Amazon Elastic Compute Cloud (EC2),** and the **Amazon Simple Storage Service (S3)**
- *Platform as a Service, PaaS*, offers runtime environments (operating systems, databases, security integration, or web service frameworks), application framework or middleware as a service in which users can build and deploy custom applications developed using programming languages and tools supported by the platform. Examples are **Microsoft Azure Engine**, **Google App Engine**, and **GroundOS (GOS)**
- *Software as a Service, SaaS*, offers end-user applications delivered as a service, rather than the traditional, on premise software. Users run the applications in the cloud, without managing the hardware infrastructure or platform. **Esri's ArcGIS Online** and **Salesforce.com CRM** are examples of SaaS clouds.

Other than IaaS, PaaS and SaaS clouds; Data as a service (DaaS) is a cloud service typically essential for Geospatial applications due to large volumes of data involved. DaaS is usually implemented within a SaaS, PaaS or IaaS solution and provides data within applications that support data discovery, access, manipulation, and use.

According to Díaz *et al* (Díaz et al., 2012), cloud computing presents two major benefits to SDIs: simplified deployment and maintenance of SDI services (thus increasing number of content offerings), and reduced costs of providing content and applications (which may increase the quality of service).SDIs stand to benefit immensely from cloud computing because of large datasets in SDIs and high computational requirements. Other benefits include (GeoConnections, 2013): demand for framework data, which is best provided as a service to SDIs, and the increasing need for high volume datasets such as resource management, agriculture, and demographics. Despite their numerous benefits, there are several concerns that needs to be considered with cloud computing (Ludwig, 2012; GeoConnections, 2013), such as security and availability, privacy, integrity and confidentiality, legal and liability and regulation.

Figure 2: Features of cloud services

Infrastructure as a Service (IaaS)	Platform as a Service (PaaS)	Software as a Service (SaaS)
Application	Application	Application
Development Tools	Development Tools	Development Tools
Runtime	Runtime	Runtime
Middleware	Middleware	Middleware
Application Server	Application Server	Application Server
Operating System	Operating System	Operating System
Networking	Networking	Networking
Storage	Storage	Storage
CPU/RAM	CPU/RAM	CPU/RAM

Customer Managed
Provider Managed

2.4.6. *Volunteered Geographic Information (VGI)*

VGI can be described as the application of user-generated content in the spatial information domain (GeoConnections, 2013). VGI typically involves an active community, playing a much more organized and influential role; for example in data collection or correction. In addition, VGI can act as a valuable mechanism to encourage public participation, thus engaging and empowering citizens (Carpenter and Snell, 2013).

User generated content can also help local communities generate and access data especially where more powerful entities (the Government and large corporations) commoditize spatial data (Williams et al., 2014), making it difficult for ordinary citizens to access. This can help alter the relative power that the traditional producers of data hold, while at the same time creating new avenues for local communities to influence change.

SDIs can benefit from VGI in several ways. Content produced by citizens is cost effective since they utilize local knowledge, skills and expertise. Furthermore, VGI offers the ability to access real-time data thus enhancing the SDI content. An active user community is one of the requirements for successful SDI, and VGI encourages user participation. VGI has been successfully used during disasters such as the Haiti's Earthquake in 2010 (Sui and Goodchild, 2011). Despite the numerous benefits, VGI presents several risks such as data quality concerns, legal issues, archival and presentation, and security.

2.4.7. *Free and Open Source Software (FOSS)*

The motivation behind free software is freedom, akin to freedom of speech and not free-of-cost advantages (Steiniger and Hunter, 2012). *Free software* grants freedoms of use, modification and redistribution of the software to the public, and *proprietary software* takes these freedoms away. The term *open source* denotes the availability of the software's source code; with a license authorising anyone to use the software, modify it, and even redistribute it.

In countries where the development of SDIs is in its early stages, the availability of FOSS solutions offers a genuine alternative to proprietary software (Carpenter and Snell, 2013). The drawback with FOSS is that the ongoing maintenance and training skills have to de developed internally, necessitating the need for higher qualified IT personnel.

FOSS used for geospatial applications may be categorised as (Steiniger and Hunter, 2012): Desktop GIS, Spatial Database Management Systems, Web Map Servers, Server GIS, Web GIS clients, Mobile GIS, Libraries, GIS Extensions, Plug-ins and APIs, Remote Sensing Software, and Exploratory Spatial Data Analysis (ESDA) software.

The open source geospatial community has a well-established arrangement through the Open Source Geospatial Foundation (OSGeo) and a vibrant user community who champion its potential (Carpenter and Snell, 2013). FOSS are experiencing an increasing level of collaboration among projects today (Steiniger and Hunter, 2012); resulting in projects such as the Java Topology Suite (JTS), Geometry Engine - Open Source (GEOS), and NetTopologySuite (NTS); as well as interoperability libraries such as Geospatial Data Abstraction Library (GDAL). Similarly, with the introduction of new OGC standards, the projects are striving for compatibility with each other.

3. METHODOLOGY

3.1. Introduction

The methodology used to demonstrate the technology trends that can enhance wider adoption of SDIs in Africa is presented, based on a simple geospatial application utilizing data from Kenya's Ministry of Education, Science and Technology (MOEST).

3.2. Description of the Methodology

3.2.1. Background
The technology trends are demonstrated using an application that analyses performance of KCPE results of 2015. MOEST datasets are chosen largely due to higher availability of data in the sector, including the school mapping project of 2007.

One of the most important considerations is choice of a cloud platform. There are several types of clouds, and service providers such as AWS and Microsoft Azure Engine. The GCS platform is chosen for several reasons. First, it is a flexible and powerful platform, facilitating services such as GCE, GKE and GAE. Secondly, it is still a relatively new platform with promising potential for geospatial research.

3.2.2. Google Container Engine (GKE)
In August 2015, the Google Container Engine (GKE), an IaaS cloud, became generally available. The GKE service offering provides standard IaaS features, and takes its offering to a higher level through Docker containers and Kubernetes. While Docker offers the lifecycle management of containers, Kubernetes provides orchestration and management of container clusters.

Docker uses the Linux Container technology to package applications into portable, hardware-isolated containers, allowing them to be moved between different platforms. Kubernetes, which means "pilot" or "helmsman" in Greek, is the open source cluster manager tool from Google that is used to manage Docker containers.

Among the attractive GKE features are: automated container management for running Docker containers; easy setup of clusters; declarative management of container resources and requirements such as the CPU/memory in a simple JavaScript Object Notation (JSON) or YAML file; and a flexible & open source environment using Kubernetes and other tools.

GKE facilitates Docker containers which make packaging applications easier; and Kubernetes is a powerful cluster manager and orchestration system necessary to bring workloads to production. When the needs of an application grows, resizing a cluster with more CPU or memory allocation using Kubernetes is very easy (Craig Mcluckie, 2015).

GKE charges a flat fee per hour per cluster for GKE's cluster management, depending on the number of nodes in the cluster. There is no charge for up to 5 Nodes, but more than 6 nodes attracts a charge of $0.15 HR/cluster. To minimise the cost, this research will be using 3 nodes.

3.2.3. Configuration of the GKE Platform

The GCS console can be accessed at https://console.cloud.google.com/. Authentication is accomplished using a standard Google Account.

The GKE platform defines the following terms:-

- A *container* is a Linux-based technology comprising isolated guest virtual machines (VMs) installed on top of the operating system's kernel, which runs on the hardware node.
- A *container cluster* consists of the specified number and type of Google Compute Engine (GCE) instances.
- A *pod* is a collection of containers that are scheduled onto the same host. It represents logical collections of containers that belong to an application. Pods serve as units of scheduling, deployment, and horizontal scaling/replication.
- A *replication controller* ensures that a specified number of pod *replicas* are running at any one time.

3.2.4. Software Considerations

GeoServer, an open-source and Java-based software, was chosen to power the server-side geospatial services. It can be used to publish geospatial data on the network using OGC standards such as WMS, WFS and WCS; and additionally Web Map Tile Service (WMTS), CSW and WPS. GeoServer functions as the reference implementation of the OGC.

Another open-source software choosen is PostGIS, a spatial database extender for PostgreSQL DBMS. PostGIS facilitates SQL queries on spatial datasets using spatial operators such as *Point in Polygon,* which facilitates the determination of the county in which a school belongs to.

The application uses a web server known as Nginx (pronounced *engine-x*). Running on a Linux platform, Nginx is an open source reverse proxy server for HTTP, HTTPS and other protocols, a load balancer, HTTP cache, and a web server.

Finally, OpenLayers was used for client-side interaction. OpenLayers is an open source JavaScript library for displaying geospatial data in web browsers. It provides an API for building web-based geoinformation applications. OpenLayers is used together with HTML to give interface to the application, as well as user interaction.

The technology stack is completely FOSS, consisting of Linux, GeoServer, PostGIS, OpenLayers, Docker and Kubernetes. Thus the application was developed without significant investment in software licensing costs. In addition, support and online user's communities are readily available, making it easy to access resources when configuring the cloud infrastructure.

3.2.5.   Installation Steps

The Google Cloud Shell, which runs a virtual machine instance of Debian Linux operating system, was activated. The shell provides a command-line access to computing resources hosted on GKE, using gcloud and kubectl utilities that are continuously used in the setup and configuration.

There are many Docker images that can be reused, and two images suitable to this research are: https://github.com/kartoza/docker-geoserver and https://hub.docker.com/r/mdillon/postgis/, representing GeoServer and PostGIS images, respectively.

The first step was to use gcloud to set up the GCE compute zone, as shown in Code 1. GCE resources live in regions or zones, a specific geographical location where a resource runs.

```
$ gcloud config set compute/zone us-central1-b
```
Code 1: Creating the GCE Compute Zone

Pulling the Docker images was accomplished by the commands shown in Code 2.

```
$ docker pull mdillon/postgis
$ docker pull kartoza/geoserver
```
Code 2: Pulling Docker Images

The next step was to create a GKE cluster with three nodes on which GeoServer, PostGIS and the Nginx webserver will run. Code 3 shows the creation of a cluster named gscont with 3 nodes.

```
$ gcloud container clusters create gscont --num-nodes 3
$ gcloud config set container/cluster gscont
$ gcloud container clusters get-credentials gscont
```
Code 3: Creating the Cluster

The application makes use of persistent disks, allowing GeoServer, PostGIS and Nginx to preserve their state across shutdown and start-up. The disks are created as shown in Code 4.

```
$ gcloud compute disks create --size 200GB postgis-disk
$ gcloud compute disks create --size 200GB geoserver-disk
```
Code 4: Creating the Disks

The final step was to create and start the pods, and the services used to access the pods. Pod and Service specifications are defined in YAML or JSON files. Creating the postgis pod and its service was accomplished by Code 5.

```
$ kubectl create -f mpostgres.yaml
$ kubectl create -f postgres-service.yaml
```
Code 5: Creating the Pods

To create an image instance of Nginx, a new Docker image was created, as outlined in Code 6. Subsequently, the steps to create a running web server instance are shown in Code 7.

```
FROM nginx
EXPOSE 80
COPY webdir /usr/share/nginx/html
```
Code 6: Nginx Dockerfile

```
$ docker build -t gcr.io/mywpstest/phd-nginx:v1 .
$ gcloud docker push gcr.io/mywpstest/phd-nginx:v1
$ kubectl run phd-nginx --image=gcr.io/mywpstest/phd-nginx:v1 --port=80
$ kubectl expose rc phd-nginx --target-port=80 --type="LoadBalancer"
```
Code 7: Creating the Nginx web server

Using kubectl (see Code 8 and the output in Code 9), GeoServer is running on 130.211.167.146; and can be accessed at: http://130.211.167.146:8080/geoserver/web. Opening this link loads the normal GeoServer login screen. Additionally, the web server running Nginx is on IP address 199.223.236.236, and can be accessed using: http://199.223.236.236/index.html.

```
$ kubectl get services gsfrontend psfrontend phd-nginx
```
Code 8: Inspecting the Services

```
$ kubectl get services
NAME          CLUSTER-IP       EXTERNAL-IP       PORT(S)     AGE
gsfrontend    10.103.241.171   130.211.167.146   8080/TCP    70d
kubernetes    10.103.240.1     <none>            443/TCP     70d
phd-nginx     10.103.244.246   199.223.236.236   80/TCP      46s
psfrontend    10.103.245.210   <none>            5432/TCP    70d
```
Code 9: Detailed Service Endpoints

3.3.   Data Sources

The KCPE 2015 results were obtained from MOEST. The Schools dataset, which contains spatial coordinates of primary schools, was obtained from the Kenya Open Data Portal[1]. The latter is an extract of data collected by MOEST in its School Mapping exercise carried out in 2007. Wards, constituencies, and counties shapefiles were obtained from the IEBC website.

Table 1 presents the population and sample for each dataset. To be included, a record in the schools dataset must be easily identifiable by name in the KCPE 2015 dataset. There are a lot of naming anomalies, duplicates, and other inconsistencies between the two datasets, which explains the relatively low sample representation of 55% and 59%. However, this is considered representative since the figures account for more than 30% of the population.

Table 1: Datasets in the Study

Dataset	Population	Sample	%	Source
Primary Schools	25121	13890	55	MOEST School Mapping, 2007
KCPE Results 2015	938738	551150	59	MOEST, 2015
Wards	1450	1444	99	IEBC
Constituencies	290	290	100	IEBC
Counties	47	47	100	IEBC

---

[1] https://www.opendata.go.ke/Education/Kenya-Primary-Schools/p452-xb7c

3.4. Data Loading

By loading the data into PostGIS database, GeoServer can be used to serve various WFS, WMS, and WCS content. Additionally, the application can be extended to provide more sophisticated geoprocessing services running extensible mechanisms like the OGC's WPS.

Loading data into the database takes two steps: generation of SQL scripts from shapefiles, using the shp2pgsql tool; followed by the data loading into tables in the PostGIS database. Code 10 shows an example. The srid 4326 refers to the new World Geodetic System (WGS 84), given that the shapefiles and school mapping data are in this reference system. The data is stored in a database named "gis".

```
$ shp2pgsql -s 4326 counties.shp counties > counties.sql
$ psql -f counties.sql gis
```
Code 10: Data load Scripts

3.5. Justification of the Methodology

As at the time of writing this paper, the KNSDI is not operational (in the sense that there was no geoportal). Ideally, such an application should have been developed within the framework of the KNSDI, by integrating with existing datasets and functionality.

4. RESULTS AND DISCUSSION

4.1. Introduction

An analysis and discussion of the results are presented in this section.

4.2. Data Portrayal Using WMS

A simple geospatial application running in the cloud has been developed to showcase several emerging trends. The application serves a number of maps depicting spatial characteristics of the dataset that are loaded, which are highlighted in Figure 3,
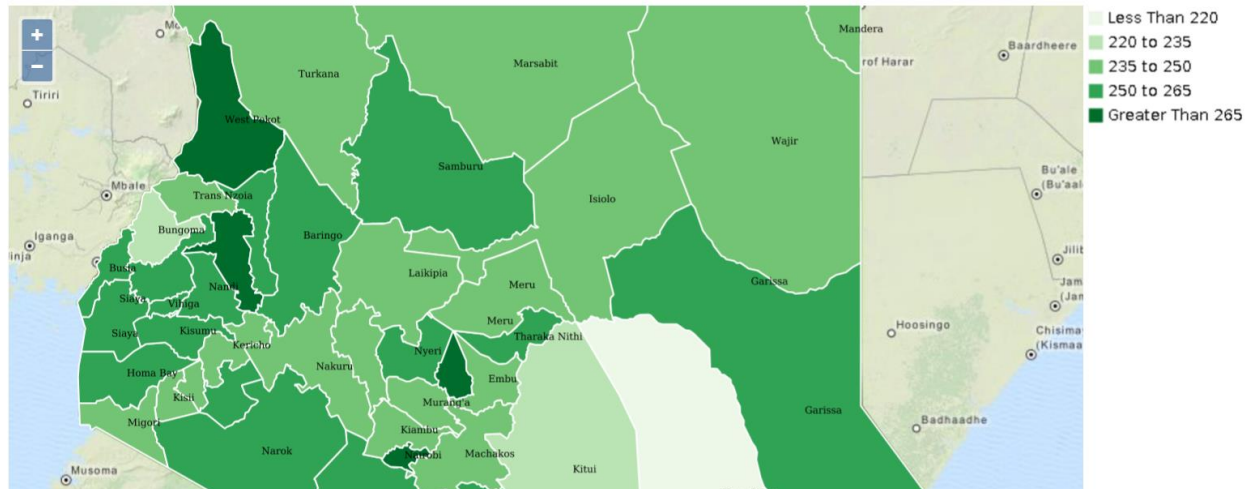
Figure **4** and Figure 5.

Figure 3: Average KCPE 2015 Results by Constituency
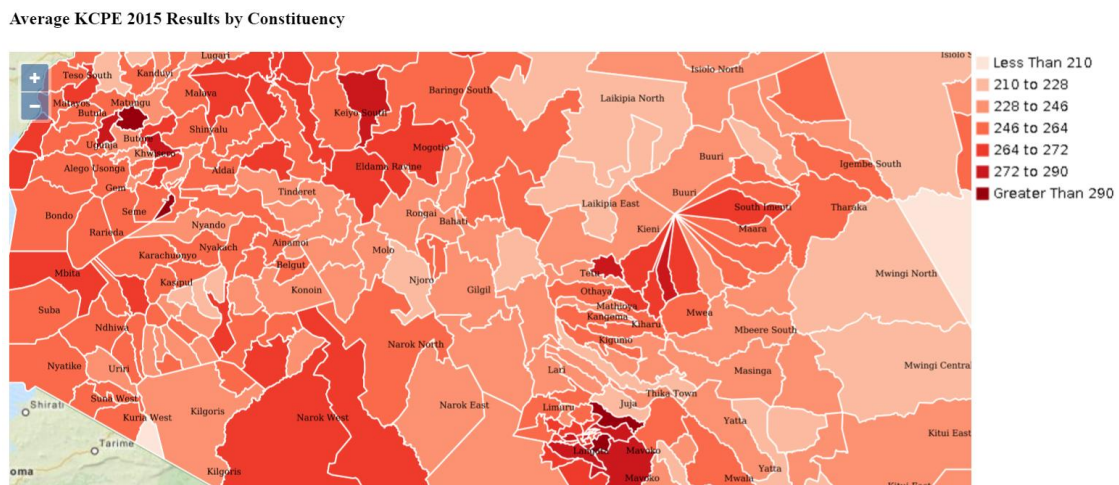
Figure 4: Average KCPE 2015 Results by County

Average KCPE 2015 Results by County

The database can be used to build and visualize complex spatial queries. For instance, one might be interested to know which counties produced the top 100 candidates, or the bottom 100. One such query, illustrated in Code 11, has been used to produce the output in Figure 5.

```
create view county_top_100 as
with query2 as ( select name, total, sch_code, name1, level, status, sponsor, type1, type2,
type3, counties.county_cod, counties.county_nam, counties.geom
        from kcpe2015, schools, counties
        where kcpe2015.total IS NOT NULL AND kcpe2015.total > 0 AND kcpe2015.sch_code =
schools.code AND schools.county = counties.county_cod order by kcpe2015.total DESC limit 100)
select query2.county_cod, query2.county_nam, query2.geom, count(query2.geom) as count
from query2 group by query2.county_cod, query2.county_nam, query2.geom
order by count; insert into gt_pk_metadata (table_schema, table_name, pk_column, pk_policy)
values ('public', 'county_top_100', 'code', 'assigned');
```
Code 11: Sample Spatial Query

Figure 5: Top 100 Pupils by County

Top 100 Pupils by County
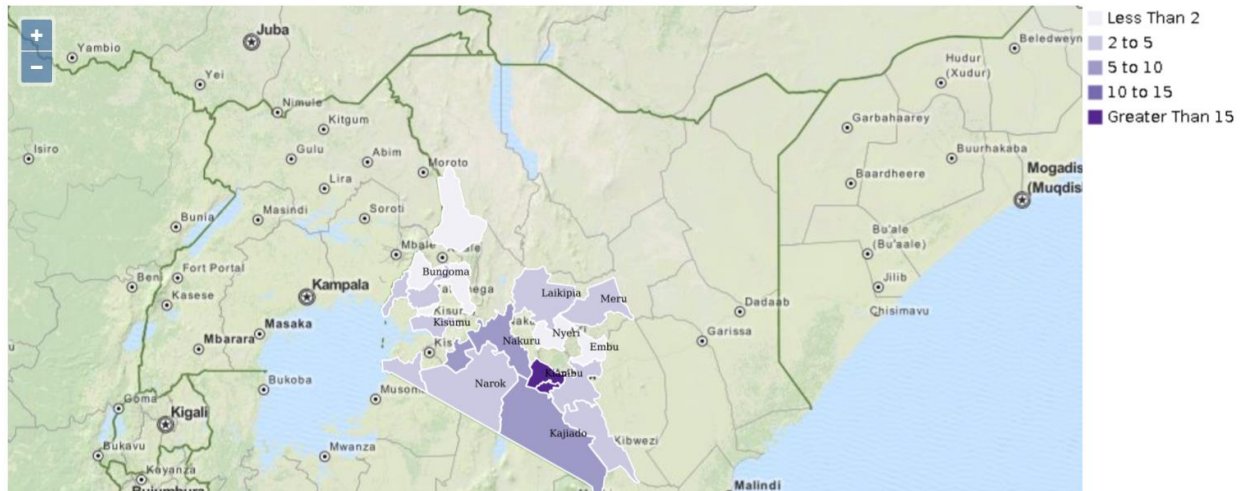
4.3. Benefits to SDIs

The application has demonstrated the potential benefits of cloud, FOSS and other trends to SDIs. The major benefit of the cloud system is flexibility and scalability where the computing resources are increased as the number of users grows. This can be used to accommodate a very large number of users, while supporting efficient geoprocessing functionality.

The application has been configured to use a custom machine type with a memory of 2GB and 1 CPU. Higher computational or memory machine types are available but attract higher charges. To serve the needs of diverse uses and applications, GCE provides various machine types including high-CPU (with up to 32 CPU and 28 GB RAM) and high-memory (with up to 32 CPU and 208 GB RAM).

4.4. Areas for Further Study

Several challenges were encountered during the study. Firstly, a significant amount of time was spent identifying records in the two datasets. This can be eliminated using Linked Data, where relevant agencies cooperatively work together to improve data management using. Additionally, although Google provides a free three month trial period, continued use of cloud resources especially GCE incurs charges. At the moment the services have been shut down but can be reactivated in case there is need for review.

The following are recommendations for further improvement.

- The application can be extended to utilize emerging mobile data collection tools, which may give heads of schools the ability to update their own school's data. Data elements such as type of school, pupil to teacher ratio, pupils to class ratio, number of classrooms, total number of pupils, and location of the school can be routinely updated. This can be regarded as a form of VGI since it supports user-driven spatial data collection.
- The application can also be extended to incorporate Linked Data. In this scenario, each school can be allocated a unique URI which identifies the school on the web, with the school's data stored in RDF format which incorporates richer data and service description.

5. CONCLUSIONS AND RECOMMENDATIONS

The application has shown that cloud computing can increase uptake of SDIs, through highly scalable web services running in the cloud. Cloud services have the potential to provide higher quality of services, improved performance, reliability of GI services, and improved accessibility to geospatial data and services. Several risks and benefits of cloud computing, which practitioners should reference as they deploy SDI in the cloud, have been discussed.

The advantage of cloud services is that the application can be scaled to accommodate a very large number of users. However, time and resources did not allow us to demonstrate the scalability of the application, but there is huge potential given the scale of the cloud infrastructure.

FOSS software, including PostGIS, GeoServer, Linux, Kubernetes, Docker, and OpenLayers, were used extensively. The application has shown that FOSS software has great potential in SDIs, particularly in developing countries where resources are limited. However, these technologies come with a steep learning curve, which can be managed through training.

Part of the data used in the study was obtained from the Kenya Open Data Portal, an important world-wide trend. Availability of readily accessible data is a catalyst for successful development of SDIs, while creating more employment opportunities, entrepreneurship through data and service provision, and an agile ecosystem of developers built around SDIs.

Other than the technology trends for SDI demonstrated throughout this paper, the application may be of interest to Geospatial engineers and practitioners in the Education sector who may want to carry out spatial analysis of the existing data. The principles discussed can be applied to other application areas, including Agriculture, Health and Climate Change.

6. REFERENCES

Baranski, B. (2008). "Grid Computing enabled Web Processing Service", *Proceedings of the 6th Geographic Information Days*, at http://gi-days.de/archive/2008/downloads/acceptedPapers/Papers/Baranski.pdf, [accessed 9 January 2016].

Béjar, R., M.Á. Latre, J. Nogueras‐Iso, P.R. Muro‐Medrano and F.J. Zarazaga‐Soria (2009). An Architectural Style for Spatial Data Infrastructures, *International Journal of Geographical Information Science*, 23(3): 271–294.

Brauner, J., T. Foerster, B. Schaeffer and B. Baranski (2009). "Towards a Research Agenda for Geoprocessing Services", *12th AGILE International Conference on Geographic Information Science 2009*, at https://agile-online.org/Conference_Paper/CDs/agile_2009/AGILE_CD/pdfs/124.pdf, [accessed 2 January 2016].

Brink, L. van den, P. Janssen, W. Quak and J. Stoter (2014). Linking spatial data: semi-automated conversion of geo-information models and GML data to RDF, *International Journal of Spatial Data Infrastructures Research*, 9(2014): 59–85.

Budhathoki, N.R. and Z. Nedović-Budić (2007). "Expanding the Spatial Data Infrastructure Knowledge Base"*Harlan Onsrud (ed), Research and Theory in Advancing Spatial Data Infrastructure Concepts*, Redlands: ESRI Press, pp. 7–31.

Carpenter, J. and J. Snell (2013). Future trends in geospatial information management: the five to ten year vision.

Craig Mcluckie (2015). *Google Container Engine is Generally Available*, at http://googlecloudplatform.blogspot.co.ke/2015/08/Google-Container-Engine-is-Generally-Available.html, [accessed 9 January 2016].

Díaz, L., A. Remke, T. Kauppinen, A. Degbelo, T. Foerster, C. Stasch, M. Rieke, B. Schaeffer, B. Baranski and A. Bröring (2012). Future SDI – Impulses from Geoinformatics Research and IT Trends, *International Journal of Spatial Data Infrastructures Research*, 7: 378–410.

Executive Office of the President (1994). Coordinating Geographic Data Acquisition and Access, the National Spatial Data Infrastructure. Federal Register 59:17671-4.

GeoConnections (2013). Spatial Data Infrastructure (SDI) Manual for the Americas.

GeoConnections (2016). Impacts and Implications of Big Data for Geomatics: Backgrounder.

Goodchild, M.F., M. Yuan and T.J. Cova (2007). Towards a general theory of geographic representation in GIS, *International Journal of Geographical Information Science*, 21(3): 239–260.

Henricksen, B. (2007). United Nations Spatial Data Infrastructure Compendium.

Li, W., L. Li, M.F. Goodchild and L. Anselin (2013). A Geospatial Cyberinfrastructure for Urban Economic Analysis and Spatial Decision-Making, *ISPRS International Journal of Geo-Information*, 2(2): 413–431.

Ludwig, B. (2012). Implications of security mechanisms and Service Level Agreements (SLAs) of Platform as a Service (PaaS) clouds for geoprocessing services, *Springer in Applied Geomatics*.

Maguire, D.J. and P.A. Longley (2004). The emergence of geoportals and their role in spatial data infrastructures, *Computers, Environment and Urban Systems*, 29(2005): 3–14.

Newman, G., D. Zimmerman, A. Crall, M. Laituri and J. Graham (2010). User-friendly web mapping: lessons from a citizen science website, *International Journal of Geographical Information Science*, 24(12): 1851–1869.

Steiniger, S. and A.J.S. Hunter (2012). The 2012 free and open source GIS software map – A guide to facilitate research, development, and adoption, *Computers, Environment and Urban Systems*, 39(2013): 136–150.

Sui, D. and M. Goodchild (2011). The convergence of GIS and social media: challenges for GIScience, *International Journal of Geographical Information Science*, 25(11): 1737–1748.

Tsinaraki, C. and S. Schade (2016). Big Data – a step change for SDI?, *International Journal of Spatial Data Infrastructures Research*, 11(2016): 09–19.

Tumba, A.G. and A. Ahmad (2014). Advocating for Spatial Data Implementation at the Lower Tiers of Governments in Developing Countries: The Case of Africa, *Universal Journal of Geoscience*, 2(5): 152–159.

Wainainah, D. (2015). Future is in Smart Homes as Internet of Things catches on.

Williams, S., E. Marcello and J.M. Klopp (2014). Towards Open Source Kenya: Creating and Sharing a GIS Database of Nairobi, *Annals of the Association of American Geographers*, 104(1): 114–130.

Yang, C., R. Raskin, M. Goodchild and M. Gahegan (2010). Geospatial Cyberinfrastructure: Past, present and future, *Computers, Environment and Urban Systems*, 34: 264–277.