Development of WebGIS Based on Service Oriented Architecture and Cloud Computing

Ibnu Daqiqil Id1, Sukamto1, and Evfi Mahdiyah1

1Study Program of Information System, Universitas Riau, Indonesia

Ibnu.daqiqil@lecturer.unri.ac.id, sukamtonursyam@gmail.com, evi.mahdiyah@lecturer.unri.ac.id

Abstract: This paper explains about development of a Web Geographical Information System (Web GIS) based on Service-Oriented Architecture (SOA) on Cloud Computing Platform. Our study focuses on building models using SOA architecture and deployment services through a private cloud-computing environment on governmental organizations. At the global level, the cloud architecture can improve the utilization of resources effectively while SOA can create data interoperability, service flexibility, ease of development, scalability and reliability.

Keywords: Cloud Computing; Geographical Information System; Service Oriented Architectural Web GIS.

1. Introduction

Nowadays, geospatial data, information and system provide an important role for government and private sector in decision-making. A GIS (Geographical Information System) is a system designed to capture, store, manipulate, analyze, manage, and present all types of geographically referenced data (geospatial data) [1]. GIS allows us to view, understand, question, interpret, and visualize data in many ways that reveal relationships, patterns, and trends in the form of maps, globes, reports, and charts [2]. In other words, GIS is the merging of statistical analysis, database technology and cartography that can integrated into any information system. Using web technology, access to geospatial data can be improved. In addition using web GIS help decision-makers to manage their assets better, enabling faster responses and improve cross-sectoral analysis and communication process [3].

Share and access geospatial data between different partners, different sources, as well as different organizations can save money and avoid data duplication (interoperability). Information exchanges can solve problems related to inconsistencies and quality differences in the geospatial data from various sources [2]. In this case there are three important barriers in front of share and access of geospatial data; non-interoperability of geospatial processing systems which prevents sharing geospatial data, insufficient message exchange patterns which restricts access to geospatial data by subset of all potential users; and large data in different formats and forms [2] [4]. Beside those barriers, there are several problems facing GIS, which challenge the management and analysis such as [5]: Restricted in size and availability; The need to be reserved for usage; Limited storage space both locally and in network; Setting up of clusters need large amount of initial investment; Failure rates are very high when webgis used by large number of people; Maintenance can be very expensive and time consuming for real-time GIS applications.

Technologies such as Service Oriented Architecture (SOA) and cloud computing can solve this problem. Thus, our study focused on building models using SOA architecture and deployment services through a private cloud-computing environment on governmental organizations. At the global level, the cloud architecture can improve the utilization of resources effectively while SOA can create data interoperability, service flexibility, ease of development, scalability and reliability. This
Development of WebGIS Based on Service Oriented Architecture and Cloud Computing

paper uses Kabupaten Bengkalis and Kabupaten Kepulauan Meranti as case study and we try to integrate these services with the services that owned by a national spatial data network.

2. GIS SERVICE ORIENTED ARCHITECTURE

SOA concept emerged as a way of defining the software architecture with a more sophisticated way by giving greater attention among major software components. Technically, SOA is the result of the natural evolution of the Component based model into Service Orientation. Service orientation retains the benefits of object oriented systems and component-based development (self-description, encapsulation, dynamic discovery and loading) [6]. The shift is in the paradigm, which has stirred from remotely invoking methods on objects, to passing messages between services. SOA is a way of designing, developing and managing systems [7]. In addition, SOA puts emphasis on reusability by separating the interface function from internal implementation. This separation makes SOA suitable to be implemented both heterogeneous and distributed architecture [6].

One of the key benefits of service orientation is loose coupling. The principle of loose coupling is that of using a resource only through its published service and not by directly addressing the implementation behind it [8]. In this way, changes in the implementation of the service provider will not affect customer service. By maintaining a consistent interface, the service consumer can choose an alternative service life at the same service without modifying their application.

The key component in the SOA is services. A service is well-defined set of actions. It is self-contained, stateless, and does not depend on the state of other services. The service consumer and provider do not have to have the same technologies for the implementation, interface, or integration when services are used [9].

The implementation of service using open standard and widely used protocols and technologies called Web services [10]. Web Services are the basic components of distributed service-oriented systems. The World Wide Web Consortium (W3C) defines Web Services as a software system designed to support machine-to-machine interaction over the Internet [11]. Each Web service has an interface described in a machine-process able format. Other systems and services interact with the Web service in a manner described by its description using messages. Messages are conveyed typically using HTTP with an XML serialization, in conjunction with other Web-related standards, but any other communication protocol can be used as well [11].

![Figure 1. Research Methodology](http://www.estech.org)

SOA has applied to the GIS domain. Open Geospatial Consortium (OGC) has created several standards for GIS. OGC is an international industry consortium of companies, government agencies and universities participating in a consensus process to develop publicly available interface specifications [12]. The Specifications support interoperable solutions that "geo-enable" the Web, wireless and location-based services, and mainstream IT. The specifications empower technology developers to make complex spatial information and services accessible and useful with all kinds of application [4].
3. Material & Methodology

The flowchart of methodology used in the present work is shown on Figure 1. There are five phases involved in the methodology. In the first step, we do business modeling. At this step, we define the interaction between the role, user and systems with use case diagram. A use-case realization represents the design perspective of a use case. It provides a construct in the design model that organizes artifacts related to the use case. This can take various forms. It can include, for example, a textual description (a document), class diagrams of subsystems and participating classes, and interaction diagrams (sequence and communication diagrams) that illustrate the flow of interactions between class and sample subsystem.

The second step of our methodology is service-oriented analysis. Service-oriented analysis consists of:

- Define Requirement. First step of service-oriented analysis related to the requirement; business needs to have a direct affiliation with the business activities and used to communicate how new or existing business processes should work. These requirements often include business process models and analysis using standard flowchart notations to depict business processes; Key business stakeholders requirements; Logical data model and data dictionary references. Defining and documenting all business requirements necessary for this step.
- Define automation system. Automated systems eliminate the need for human interference in order to complete a task. Several industries use automated systems to increase production and reduce costs. Automated systems used to handle a wide range of tasks.
- Explore process automation; explore what portions of the business processes can and automated. Understand how any existing systems often referred to as legacy systems, fit into the organization and derive system requirements. The business model will serve as a model system.
- Identify candidate services. Candidate service is a service that meets the essential requirements. The service candidate term is used help distinguish a conceptualized service from an actual implemented service. This distinction is especially important when documenting service inventories as part of blueprint specifications or even when keeping track of a service's progress via its service profile. In some cases, candidates may have application services to more than one important requirement.
- Decomposes business process and review the details. Decompose business process is the process of breaking complex business process into smaller sub-parts, and then breaking those smaller parts down even more, until the complex business process has been broken down into more discreet components with a more understandable structure.
- Last step is determines category of each services. These categories are three cases: business category, application category, and GIS category. Each service decomposes to subservices and in final step; each subservice decomposes to one or more atomic service.

![Figure 2. Service Oriented Analysis Step](image)

Third step is service-oriented design. Service-oriented design is about creating systems that group functionality around logical function and business practices. The goal of the service-oriented design is to separate parts of the application or system into components that can be repeated in, enhanced, and do not need to test and verify all other components when there are changes to certain components. Achieving these goals usually entails a trade-off between complexity and iteration speed.
4. Results and Discussion

4.1 Cloud Architecture

There are some key considerations in designing cloud architecture for GIS applications such as performance, scalability, availability, reliability, cost, and ease of management. In terms of scalability, availability and reliability of used Load balancers and separation between caches, application and database servers. Load balancers used to improve performance and reliability by distributing the workload across multiple servers. If one of the servers that is load balanced fails, the other servers will handle the incoming traffic until the failed server becomes healthy again. Load balancer also can be used to serve multiple applications through the same domain and port, by using a layer 7 (application layer) reverse proxy.

![Figure 3. Infrastructure Design](image)

Besides load balancer, application and database tiers do not contend for the same server resources (CPU, Memory, I/O, etc.). Separation of database management system (DBMS) can eliminate the resource contention between the applications. One way to improve performance of a database system that performs many reads compared to writes, such as a Map Service and geo-processing and image-processing service is use master-slave database replication and setup cache server.

Master-slave replication requires a master and one or more slave nodes. With Master-slave configuration, we can improve availability, performance and data consistency. If one master fails, other masters will continue to update the database. Masters can be located in several physical sites or distributed across the network. All updates sent to the master server/node and reads distributed across all nodes. This strategy can improve database performance, availability and scalability. We can vertically scale each tier separately, by adding more resources to whichever server needs increased capacity. To increase security, we remove the database from the DMZ, or public internet.

4.2 GIS Architecture

GIS Architecture consists of two layer: Service Layer and Data layer. Service layer is responsible for processing requests spatial data, finding data or services from registries, and store spatial data. In addition, the service layer also provides functionality to authentication and authorization, data management, error management and application management.

![Figure 4. Web GIS Architecture](image)
Development of WebGIS Based on Service Oriented Architecture and Cloud Computing

Service layer consist of service handler, OGC Interface, catalog services, map services, geoprocessing services, image processing service, session management, error handling, authorization, reporting and analysis. Data layer is layer that responsible for storing, update and fetch data from service layer. This layer consists of meta data repository (a database created to store metadata), datasets, model repository and databases system. The separation of service and data layer to improve portability.

Service handler act as connection points from the listener/service consumer to another webgis services. OGC Catalogue interface standards specify service interfaces, data bindings, and defining application profiles required to publish and access digital catalogues of metadata for geospatial data, services. Catalogue services provide function to publish and search collections of metadata (descriptive information about data) for data, services, and other related information objects. Those metadata stored in Metadata Repository. This repository act like database system, can query and present data for evaluation and further processing. Catalogue services are required to support the discovery and binding to registered information resources within an information community.

Web Map Service provides HTTP interface for requesting geo-registered map images from one or more distributed geospatial databases. The response to the request is one or more map images (returned as JPEG, PNG, etc) that can be displayed a browser and desktop applications. Web Map Service also known as Web Map Service Interface Standard (WMS). The WMS standard defines three method:

- GetCapabilities. Obtain service-level metadata, which is a machine-readable and human-readable description of the WMS’s information content and acceptable request parameters.
- GetMap. Obtain a map image whose geospatial and dimensional parameters are well defined.
- GetFeatureInfo. Ask for information about particular features shown on a map.

This operations used in HTTP is requested by users. Users do twice request to get the first map is a request to obtain metadata pedta and request the map layer (Figure 6). Figure 6 describe example of detail interaction between user and WMS server. Each operation produces a response in a particular format. WMS, which supports SLD (Styled Layer Descriptor) generally, has DescribeLayer and GetLegendGraphic operations.

**Figure 5. Service Output Example**

4.3 **Research Output**

There are two main output of this research, service layer and geoportal. Service layer is service that handle request from geo portal, another gis client, another GIS server and National Spatial Data
Development of WebGIS Based on Service Oriented Architecture and Cloud Computing

Network. This service in accordance with the OGC. One sample can be seen in Figure 7. Besides service layer, other outputs of this research is the geo portal. On the portal, there is a geospatial data from various sources and data from different live map sources. In another word Geo portal id a centralized and uniform interface to access the distributed and heterogeneous resources and services. The portal views can be seen in Figure 6.

5. Conclusion

In this paper, the implementation of webgis using SOA architecture over cloud computing flatom was demonstrated. The proposed solution provides geospatial data through geospatial Web services using various kinds of message exchange patterns for different kinds of clients. With cloud computing we can easily scale the infrastructure. However, this environment still has two single points of failure (load balancer and master database server), but it provides all of the other reliability and performance benefits that were described in each section above.

References
