

# SOA Based Generic Payroll Management System

Zaid Abdulkadir

HiLCoE, Computer Science Programme, Ethiopia  
zaidabdulkadir@gmail.com

Fekade Getahun

HiLCoE, Ethiopia  
Department of Computer Science, Addis Ababa  
University, Ethiopia  
fekade.getahun@aau.edu.et

---

## Abstract

Service based software acquisition is one of the promises of Service Oriented approach. Combination of SoA and Cloud based deployment reduces cost of environment (i.e., hardware and software) and operating cost. In this regard, SOA Based Generic Payroll Management System that takes into account different domain variability is developed. Using Software as a service, users can use either the default configuration or they can customize their preferred service.

The main variability in a payroll system is income tax formulation which depends on the rule/regulation of a country that uses salary range, age and/or gender as determinants. In this work, we developed a prototype that maintains relevant entities such as person, employee, department, payment and variabilities using column-based data model.

**Keywords:** Generic Payroll; Cloud Based System; Service Based Payroll; SOA

---

## 1. Introduction

Service-oriented Architecture (SOA) is an emerging style for architecting applications. It promises to better align IT with business processes. In reality, users still play a central role in today's business processes and generation of user interfaces [1]. SOA is a blueprint for an adaptable, flexible, and open IT architecture and usable for developing solutions. Easy adoptability of SOA enables companies in diverse industries to quickly differentiate their businesses and optimize processes in their business networks.

Recently systems are developed for the cloud environment [2]. Personalization of features, configuration options, customizable screens, personal menus, customizable settings and reports, and flexibility for end-user are main notions considered in custom programming. Most modern payroll systems are also web-based, running from zero footprint on client browsers. This paper deals on the Design and Development of a Generic Web Based Payroll Management System using SOA. The prototype is deployed on a cloud environment. Cloud computing

represents a form of sharing resources and as such is a basic form of collaboration.

Enabling the information exchange using typical cloud service concepts, i.e., concepts for storage and computation services in the cloud, is one of the main usage scenarios discussed in this paper.

## 2. Related Work

Several works have been done in the area of Payroll Management System, cloud and SOA. Each work differs from the other on the set objective to be met and the type of service that is intended to be provided. The authors in [3] focused on the cloud, cloud computing and its myriad applications. They described various cloud computing paradigms and types of clouds that include public cloud, private cloud, community cloud, and hybrid cloud. A cloud computing architecture has been described with existing features. Finally, the authors deployed a sample dummy system on public cloud computing system and showed the feature that will be faced when at the time of usage on Microsoft Azure like performance, security and scalability that

differentiates it with the existing cloud computing process.

In [4], the authors worked on cloud computing for educational system and more specifically on the security issues of public clouds. At the end, the authors provided a guideline to develop a cloud computing system for education and suggested few issues to be considered in developing an educational cloud computing system including working principles, significance of development, growth of the system and implications for education.

A mathematical model for cloud computing is presented in [5] where the authors implemented a mathematical model based on market oriented cloud computing. An algorithm is also implemented from the mathematical model. The authors tested the proposed system setting up a private cloud based on Ubuntu's 10.04 Server edition. The test result shows that as stochastic demand gradually increases, the proposed objective function slowly increases and provides more optimized value when the stochastic demand is within a moderate to maximum range although didn't find best performance when maximum stochastic demand holds.

In related studies, the customer relationship management system is analyzed and discussed and a solution for customer relationship management based on service oriented architecture is presented. Some of these requirements include: business management (prices, sales, deals and include stables of prices, discounts, brand, product type) and sales management (to ensure getting the latest sales information at any time).[6] The main modules implemented are a service contract management, contract support and sales staff awareness, and sales management services to manage employee placement, salaries. The Service Management module accepts various requests from customers and classifies according to the priority of their requests and needs, and provides services related deals. Management of customer data is preliminary.. Decision Support identifies valuable customers based on their classification, degree of satisfaction and loyalty deals. Service providers includes a library of

modules thereby access to the services is possible. Since the service providers and their customers are not directly interacting with each other, the service-oriented architecture is a logical component and acts as a mediator [6].

Service-oriented CRM for Insurance organizations were presented in [6]. Some of the requirements include:

- CRM systems have an integrated sales management and service platform for organizations to offer health insurance. Resources need to be integrated and CRM platform administrator can completely control the sales process.
- CRM system for insurance agencies should be based on an open framework, and the ability to reconfigure business processes and customer service based on demand.

The system is composed of several layers such as the resource layer, business layer, presentation layer and the data model layer that supports CRM operations.

Business process layer includes operational processes, organization, insurance business, which includes sales automation processes, automate marketing processes and service after sale. Finally, it includes the presentation layer of the system, CRM system and user interface.

### 3. The Proposed Solution

The proposed system is designed using SOA based generic payroll management system that considers variability related to income tax of different countries' rule. The proposed system has two interrelated packages: one for managing user side operations and the other for payroll operations. In general, it gives a full privilege for payroll administrators. Payroll users use the payroll system to manage employee and payment information.

Service-oriented architecture is an architectural software concept that defines the use of services to support business requirements. In SOA, resources are

made available to other participants in the network as independent services that are accessed in a standardized way. Most definitions of SOA identify the use of the web in its implementation. However, it is possible to implement SOA using any service-based technology.

Unlike traditional object-oriented architectures, SOAs are comprised of loosely joined, highly interoperable business services. As these services are interoperable via different development technologies (such as Java and .NET), the software components are very reusable.

At the highest and most abstract level, the logical architectural view of this system is composed of a set of cooperating components grouped into layers. Figure 1 shows a simplified, high level representation of system components and their relationships with users, other applications that call services implemented within the application's business layer, data sources. Web services provide access to data, and external or remote services that are consumed by the application.

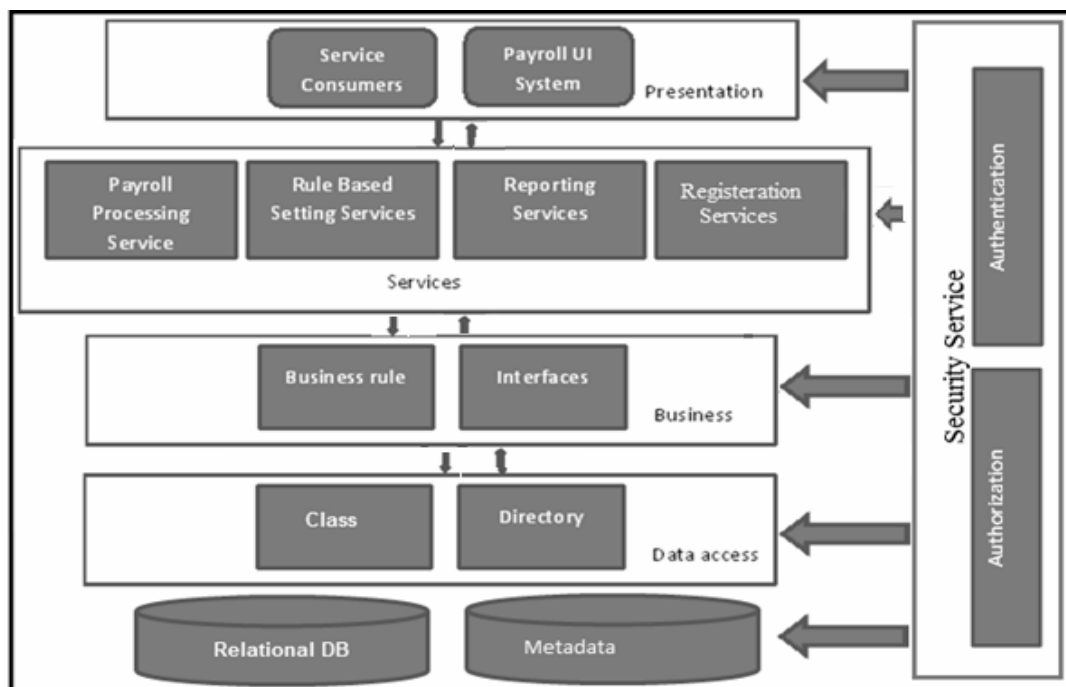


Figure 1: General Architecture

As shown in Figure 1, an application can consist of a number of basic layers. The common three-layer design shown in Figure 1 consists of the following layers.

**Presentation Layer:** contains the user oriented functionality responsible for managing user interaction with the system, and generally consists of components that provide a common bridge into the core business logic encapsulated in the business layer.

**Business Layer:** manages communication between presentation and a data access layer. The main components of the business layer are business rules

and workflows. A business rule describes a specific procedure; a workflow consists of the tasks, procedural steps, required input and output information, and tools needed for each step of that procedure. Business logic describes the sequence of operations associated to data in a database to carry out the business rule.

**Data Access Layer:** provides access to data hosted within the boundaries of the system, and data exposed by other networked systems; perhaps accessed through services. The data access layer exposes

generic interfaces that the components in the business layer can consume.

*Service Layer:* an application provides services to other applications as well as implementing features to support clients directly. The service layer effectively provides an alternative view that allows clients to use a different channel to access the application.

A user accesses the application through the presentation layer, which communicates either directly with the components in the business layer or through an application façade in the business layer if the communication methods require composition of functionality. Meanwhile, external clients and other systems can access the application and make use of its functionality by communicating with the business layer through service interfaces. This allows the application to better support multiple client types, and promotes re-use and higher level composition of functionality across applications.

#### 4. Discussion

The finding of this research basically covers the development of a SOA Based Generic Payroll Management System, a rule based tax setting design which allows users to enter their respective country or company rules.

The outcome of this research is creating generic payroll system and in addition it uses SOA to interact with modules or to communicate with third party applications.

The implementation of the prototype was taken care of using the following technologies.

- .Net framework for Web Developers.
- Asp.net web framework.
- SOA using WCF.
- Microsoft SQL Server relational database.

The documentation of the architectural analysis was put together using the Microsoft Architecture diagram methodology's documentation strategy. It consisted of laying down architecturally significant (driving) requirements, defining the project scope, showing the notional architecture using use cases, and component diagrams, analyzing the architecture

comparing it with candidate architectures. As platform and deployment environment MS Azure has been selected and presented to show the environment practically.

#### 5. Conclusion and Future Work

In general, building service oriented integrations around existing payroll solutions can be seen as an incremental approach for service enabling the organization. Another approach for service enabling an organization is to integrate different services/payroll modules from different vendors to custom built a payroll solution for the organization. This is a more aggressive and complex approach but brings more flexibility to the organization in selecting best matching payroll modules for the organization. This approach is more suitable if the organization already owns discrete IT systems that are well functional within its departments. By employing a suitable SOA enabler pattern such as 'service broker' or 'web service facades' the organization can build its unique payroll solution that leverages on most of the existing investments.

The developed system provides very user-friendly interfaces; facilitates tax rule setting, organization predefined settings and able to determine the payroll period to process. However, for future the system can be processed at any time without the defined period; In addition to this integration to human resource and finance is an important topic for future research.

#### References

- [1] Daniel Lübke, Tim Luecke, Kurt Schneider, and Jorge Marx Gómez, "Using EPCs for Model-Driven Development of Business Applications." In Franz Lehner, Holger Nösekabel, and Peter Kleinschmidt, editors, Multikonferenz Wirtschaftsinformatik 2006, Vol. 2, pp. 265–280, GITO Verlag, 2006.
- [2] Peter Mell and Tim Grance, "The NIST Definition of Cloud Computing", Version 15, National Institute of Standards and Technology (NIST), [www.csrc.nist.gov](http://www.csrc.nist.gov), 2009, Last accessed on January 2014.

- [3] N. Ram Ganga Charan, S. Tirupati Rao, P.V.S Srinivas, "Deploying an Application on the Cloud", *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 5, 2014.
- [4] P. Shanthi Bala, "Intensification of Educational Cloud Computing and Crisis of Data Security in Public Cloud, *International Journal on Computer Science and Engineering*, Vol. 02, No. 03, pp. 741-745, 2010.
- [5] K. Mukherjee and G. Sahoo, "Development of Mathematical Model for Market-Oriented Cloud Computing", *International Journal of Computer Applications*, Vol. 9, No.11, 2010.
- [6] P. Jaeger, J. Lin, J. Grimes, and S. Simmons, "Where is the cloud? Geography, Economics, Environment, and Jurisdiction in Cloud Computing," *First Monday*, Vol. 14, 2009.