SOA and Cloud Computing Approaches to Distributed Applications Using WCF

Y. KIRAN KUMAR1, J.MURALI2, K.TASLEEMA SULTHANA3

1,2,3Asst. Professor, Dept of MCA, SVEC, A. Rangampet, Tirupati.

ABSTRACT

In this paper aims to develop the distributed application system with the help of SOA Architecture and Cloud Computing so that end user connect and use the different systems which are distributed over different topographical regions. Architecture-based approaches have been promoted as a means of streamlining the complexity of system building and advancement. Thus, SOA acts as an application topology in which the business logic of the application is planned in modules (services) with clear uniqueness, principle and programmatic access across boundaries. SOA offering that provides developers, architects and IT professionals with the tools, technologies, framework and guidance for building and maintaining SOA solutions, as well as the personal productivity software that enables business users to streamline and optimize business processes in a manner that increases productivity, lowers costs, and promotes organizational agility. Windows Communication Foundation (WCF) is Microsoft’s unified framework for building reliable, secure, transacted, and interoperable distributed applications. WCF represents a new step in distributed programming for developers using the .NET Framework. Cloud computing depict wide-ranging advancement towards the operation of wide area networks, such as the Internet, to facilitate interface between information technology service supplier and clients. Cloud computing has a numeral profits and threats that should be looked at, by any higher-ranking guidance group taking into account the relocation of its enterprise computing IT portfolio. SOA and cloud computing are complementary activities and both will play important roles in IT planning.

Keywords: Service Oriented Architecture, Cloud Computing, Distributed Application, Windows Communication Foundation, Web Services, Model View Controller.

1. INTRODUCTION

Service Oriented Architecture (SOA) is a design approach to organizing existing IT assets such that the heterogeneous array of distributed, complex systems and applications can be transformed into a network of integrated, simplified and highly flexible resources. Service-Oriented Architecture (SOA) is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. A well-executed SOA project aligns IT resources more directly with business goals, helping organizations to build stronger connections with customers and suppliers, providing more accurate and more readily available business intelligence with which to make better decisions, and helping businesses streamline business processes and information sharing for improved employee productivity. The net result is an increase in organization agility. SOA is architecture and thus it is the orderly arrangement of systems that best serve the service needs of the business. Enterprise IT can succeed with SOA. However, most often it does not succeed, and much of that failure occurs because SOA implementers view SOA as something other than architecture, and most often those implementers are not architects. An SOA is a strategic framework of technology that allows all interested systems, inside and outside of an organization, to expose and access well-defined services, and information bound to those services, that may be further abstracted to process layers and composite applications for solution development. In essence, SOA adds the agility aspect to architecture, allowing us to deal with system changes using a configuration layer rather than constantly having to redevelop these systems.

2. THE PRIMARY BENEFITS OF AN SOA

- Reuse of services and behaviors, or the ability to control application behavior from application to application without a significant amount of recoding or integration.
- Agility, or the ability to change business processes on top of existing services and information flows, quickly and as needed, to support a changing business.
- Monitoring or the ability to monitor points of information and points of service, in real time, to determine the well-being of an enterprise.
- Extended reach or the ability to expose certain enterprise processes to other external entities for the purpose of inter-enterprise collaboration or shared processes.
- SOA can be used as a key technology-enabling approach to influence cloud computing
- SOA provides Loose Coupling – A component which is loosely coupled with the other components in the system makes it reusable and easy to maintain.
- SOA supports Interoperability – Ability to operate in different environment.
3. SERVICE ORIENTED ARCHITECTURE (SOA)

Now days where every organization uses different systems and technologies to develop software applications, it is a big challenge to integrate those applications together as well as with the existing legacy systems. SOA provides a great solution to create applications that leverage the different technologies and allow applications to expose their functionality via standard, interoperable interfaces.

SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments. The concept of service is not new, but the ways to implement it have been evolved over the last few years. SOA is a new way of building applications that have the following key characteristics:

- Services are components that have contracts/interfaces which are language and platform independent.
- XML and SOAP are the technologies for SOA as they are platform independent. Once these contracts are published to the outer world, consumers can dynamically discover the service and consume it.

The main objective behind SOA implementation is to break the system with the right degree of modularity and to establish the loose coupling between them. This can be achieved even without using the web service such as you can build an application where the presentation layer (for e.g. JSP) is not tightly integrated with the Data Model and can be accessed appropriately with EJBs. Web services give us, are platform independent standards as HTTP, SOAP, UDDI and XML so provides interoperability between two entirely different applications developed on two heterogeneous technologies like .NET and J2EE.

4. THE MICROSOFT SOA PLATFORM

Service Oriented Architecture is built into every aspect of the Microsoft technology stack, from the developer tools that build Web services such as .NET, to server products (like BizTalk Server and Microsoft Office SharePoint Server) which further Web service construction by connecting and orchestrating services, and finally to the composite applications that consume Web services such as web-based applications available via the Intra-, Extra- or Internet.

Using .Net to build technology architecture around the service oriented architectural model with Windows Azure as the Cloud Computing Platform. Cloud computing is a solution, defining the solution requires an architecture, and that architectural approach is SOA. Windows Communication Foundation provides a unified approach to create and consume services; BizTalk Server takes an integration-oriented approach to the problem, while the Windows Azure platform brings services to the cloud. What is common to all of these technologies is the .NET Framework. SOA will change your enterprise’s IT architecture and pave the way for cloud computing by enabling a loosely coupled architecture that allows a more efficient use of cloud resources.

WCF is a tool often used to implement and deploy a service-oriented architecture (SOA). It is designed using service-oriented architecture principles to support distributed computing where services have remote consumers. Clients can consume multiple services. Services can be consumed by multiple clients. Services are loosely coupled to each other. Services typically have a WSDL interface (Web Services Description Language) that any WCF client can use to consume the service, regardless of which platform the service is hosted on. WCF implements many advanced Web services (WS) standards such as WS-Addressing, WS-Reliable Messaging and WS-Security. With the release of .NET Framework 4.0, WCF also provides RSS Syndication Services, WS-Discovery, routing and better support for REST services.

Windows Communication Foundation is Microsoft’s next-generation Web services technology that makes it easier to connect systems and applications within the organization and across geographically distributed sites. It is the first programming model built from the ground up to provide implicit service-oriented application development. WCF takes Web services to the next level by providing developers with a highly productive framework for building secure and reliable Web service applications that interoperate across platforms. Developers can now focus on business logic and leave the underlying plumbing to WCF. In addition to...
providing a programming model for building Web services, WCF ships with a set of tools and management features that make it easier for IT Professionals to create, deploy, configure, and monitor Web services.

**Fig 1.** SOA with Microsoft .NET Framework to Microsoft Cloud AZURE Platform

### 5. SERVICE ORIENTED ARCHITECTURE (SOA) AND WCF

**Fig 2.** Service Oriented Architecture (SOA)

On the left side is the grouping for the services. This can range from 3rd party services like Google Maps, Twitter, Facebook API’s etc. To your own custom services like traditional web-services, or WCF services. In the middle we have the cloud, the network or internet connection. On the right side we have the consumers of our services (Applications).

#### 5.1 Solution Architecture

**Fig 3.** Solution Architecture

We have some pre-defined services that the framework will support. Namely a **Navigation Service** and a **Security Service**. The services are connected to one common database. However, this can be split into more modular databases, one for each service. These two services expose functions for the retrieval of information used for Navigation menus in the application(s), and the security that oversees access to all the applications (things like ‘does the user have access’, what the user sees, what actions they can and cannot perform etc). All applications developed with this framework will reside here on the right-side of the diagram.

**Fig 4.** Service Architecture
5.2 Service Architecture
Using a basic tiered-design, with a separation of concerns for each project. Each WCF Service for the framework is built with a minimum of 3 projects:
- **Entities** - This project holds the `plain old CLR objects or DTOs or a combination of both.
- **Managers** — The main driver class for each project. All communication with the repository and all business logic reside here within the manager object.
- **Repository** – Classes that provide access to data sources (databases, file system etc)

To access the Navigation Manager’s class method, the request is received from across the wire (over the network or internet connection) to the hosted WCF Service. The hosted WCF Service method is then in charge of instantiating a Manager Class (in our case, the Navigation Manager), and then calling the respective method of the Manager object.

The Manager object then performs the necessary business logic and if any data access is required, the manager class does this by instantiating/calling repository class methods. Entity objects are passed back and forth, soon transformed into their XML or JSON representation before the data is sent back to the consumer in the form of a response.

5.3 Application Architecture
MVC is the technology used to develop the applications. The entry point to the application comes from the CONTROLLER classes. Controllers are responsible for loading the VIEWs, by generating MODEL objects for the View object to interpret.

The MVC Controllers access the custom business logic via the Manager classes. The bulk of the business logic are coded in the Manager classes. Data retrieval is performed by the Repository, and Entities are passed back through to the Controller via the Manager classes. The controller then populates a Model object for the corresponding View, allowing the View to interpret it and to render the UI. The Custom Project section holds the core business logic of the application. This simply provides a separation of concerns from the MVC design of model, views and controllers.

When the application loads, a Navigation controller is created. The Navigation controller is responsible for contacting the WCF Navigation Service to retrieve the navigation data. Once data is retrieved from the service, the controller transforms this data into a MODEL object, and passes this model object to the view object for it to render a html response.

5.4 Programming Lifecycle
- Define the service contract. A service contract specifies the signature of a service, the data it exchanges, and other contractually required data.
- Implement the contract. To implement a service contract, create a class that implements the contract and specify custom behaviors that the runtime should have.
- Configure the service by specifying endpoints and other behavior information.
- Host the service.

**IService.cs (Sample Code)**

```csharp
[ServiceContract]
public interface IService
{
    [OperationContract]
    List<CustomerDetails> GetCustomerDetails(string CustomerName);
    [OperationContract]
    string InsertCustomerDetails(CustomerDetails customerInfo);
}
```

```csharp
[DataContract]
public class CustomerDetails
{
    [DataMember]
    public string CustomerName { get; set; }
    public string CustomerDetails { get; set; }
    public string CustomerPhoto { get; set; }
}
```
public string CustomerName
{
    get { return CustomerName; }
    set { CustomerName = value; }
}

SERVICE.cs
public class Service : IService
{
    SqlConnection con = new SqlConnection("Data Source=Kiran;Initial Catalog=Register;User ID=sa;Password=india");
    public List<CustomerDetails> GetCustomerDetails(string CutomerName)
    {
        List<CustomerDetails> CustomerDetails = new List<CustomerDetails>();
        con.Open();
        SqlCommand cmd = new SqlCommand("select * from CustomerInfo where CustomerName Like 
" + "@Name+'%'", con);
        
    }

    public string InsertCustomerDetails(CustomerDetails customerInfo)
    {
        cmd.Parameters.AddWithValue("@Name", customerInfo.CutomerName);
    }
}

Insert, Update and Delete Data in GridView Using WCF Service.

Now click on the Button. Data will be saved in the database table and also displayed in the GridView on the form.

Now replace UserName with Name Kiran and email id with kiran@gmail.com and click on the update Button. The updated data will be displayed in the GridView and also will be updated in the database table.

6. SOA AND CLOUD COMPUTING
Cloud computing is any IT resource, including storage, database, application development, application services, and so on, that exists outside of the firewall that may be leveraged by enterprise IT over the Internet. The core idea behind cloud computing is that it is much cheaper to leverage these resources as services, paying as you go and as you need them, than it is to buy more hardware and software for the data center. Cloud computing allows you to expand and contract your costs in direct proportion to your needs. Moreover, it shifts some of the risk around expanding your IT resources from the enterprise to the cloud computing provider.

6.1 SOA is important to cloud computing for a few key reasons:
- It is a good approach to architecture that deals with the proper formation of the information systems using mechanisms that make them work and play well together, inside and outside of the enterprise.
In order to take advantage of cloud computing, you need interfaces and architectures that can reach out and touch cloud computing resources.

**Businesses successful with SOA tend to describe SOA in the following ways:**

- SOA is about the interactions and relationships between consumers and producers.
- Defining the appropriate granularity for a service is often the most difficult and daunting task.
- The service fulfills a business need.
- Businesses successful in adopting cloud computing tend to describe their efforts as follows:
  - We can fulfill a business need quickly without the overhead of acquiring a similar system managed internally.
  - We can pay for only what we use and only acquire what we need.
  - Switching to the cloud-based solution saved money.

SOA’s success is representative of the service provider’s perspective in building and delivering a service, while cloud computing success is representative of the consumer’s perspective, indicating that cloud computing success may very well be reliant upon concepts like SOA to deliver quality services. Thus, the key for SOA practitioners is to deliver usable, reliable business services with a focus that goes beyond development.

**CONCLUSION**

SOA and cloud computing have common competencies and should be building on the experiences each provides. SOA raises IT awareness of how to deliver services, not just applications, but falls short of offering mature service level agreements to back up their efforts. SOA focuses on the problem of making systems integration more efficient, and if systems integration as a trend continues to increase as described, efficiency in this task will become increasingly important to Federal leadership teams. SOA implementation technologies such as the group of Web Service standards, Windows Communication Foundation, allows a consumer software application to invoke services across a common network. Further, they allow integration across a variety of development languages and platforms, providing a language neutral software layer. A key benefit of enterprise SOA efforts is the ability to make system-to-system interfaces consistent in the enterprise architecture, thus saving resources on future integration and hopefully improving the speed at which integration can occur or organizational agility. The emphasis of cloud computing is to leverage the network to outsource IT functions across the entire stack. SOA and cloud computing should be influencing each other to ensure success in the enterprise. Cloud computing, due to its data center and operational foundation, bring a level of maturity that is lacking in SOA.

**REFERENCES**

3. [http://nullablecode.com/2013/03/soa-architecture-wcf-mvc/](http://nullablecode.com/2013/03/soa-architecture-wcf-mvc/)

**AUTHOR BIOGRAPHY**

Mr. Y. Kiran Kumar is an Assistant Professor in the Department of Master of Computer Applications of Sree Vidyanikethan Engineering College, A.Rangampet, Tirupati, India. He is having more than 5 years of teaching experience. His areas of interests include Web Technologies, Service Oriented Architecture and Cloud Computing. He has published several national and international research papers in conferences.

Mr. J. Murali is an Assistant Professor in the Department of Master of Computer Applications of Sree Vidyanikethan Engineering College, A.Rangampet, Tirupati, India. He is having more than 5 years of teaching experience. His areas of interests include Web Technologies, Distributed database Systems, and Cloud Computing. He has published several national research papers in conferences.

Ms. K. Tasleema Sulthana is an Assistant Professor in the Department of Master of Computer Applications of Sree Vidyanikethan Engineering College, A.Rangampet, Tirupati, India. She is having more than 3 years of teaching experience. Her areas of interests include Computer Networks, Distributed database Systems.