[Unit 3: Building Cloud Networks]

Cloud Computing (CSC-458)

Jagdish Bhatta

Central Department of Computer Science & Information Technology

Tribhuvan University

Evolution from Managed service providers (MSP) to Cloud Computing:

Single Purpose architectures to multi-purpose architectures:

Data center virtualization:

Cloud data center

Service Oriented Architectures (SOA):

A cloud has some key characteristics: elasticity, self-service provisioning, standards based interfaces, and pay as you go. This type of functionality has to be engineered into the software. To accomplish this type of engineering requires that the foundation for the cloud be well designed and well architected. What about cloud architecture makes this approach possible? The fact is that the services and structure behind the cloud should be based on a modular architectural approach. A modular, component-based architecture enables flexibility and reuse. **A *Service Oriented Architecture (SOA)* is what lies beneath this flexibility.**

SOA is much more than a technological approach and methodology for creating IT systems. It's also a *business* approach and methodology. Companies have used the principles of SOA to deepen the understanding between the business and IT and to help business adapt to change.

One of the key benefits of a service oriented approach is that software is designed to reflect best practices and business processes instead of making the business operate according to the rigid structure of a technical environment. A service-oriented architecture is essentially a collection of services. A service is, in essence, a function that is well defined, self-contained, and does not depend on the context or state of other services. Services most often reflect logical business activities. Some means of connecting services to each other is needed, so services communicate with each other, have an interface, and are message-oriented. The communication between services may involve simple data passing or may require two or more services coordinating an activity. The services generally communicate using standard protocols, which allows for broad interoperability. SOA encompasses legacy systems and processes, so the effectiveness of existing investments is preserved. New services can be added or created without affecting existing services. Service-oriented architectures are not new. **The first service-oriented architectures are usually considered to be the Distributed Component Object Model (DCOM) or Object Request Brokers (ORBs), which were based on the Common Object Requesting Broker Architecture (CORBA) specification.** The introduction of SOA provides a platform for technology and business units to meet business requirements of the modern enterprise. With SOA, your organization can use existing application systems to a greater extent and may respond faster to change requests. These benefits are attributed to several critical elements of SOA:

1. Free-standing, independent components
2. Combined by loose coupling
3. Message (XML)-based instead of API-based
4. Physical location, etc., not important

Combining Cloud and SOA:

Cloud services benefit the business by taking the best practices and business process focus of SOA to the next level. These benefits apply to both cloud service providers and cloud service users. Cloud service providers need to architect solutions by using a service-oriented approach to deliver services with the expected levels of elasticity and scalability. Companies that architect and govern business processes with reusable service-oriented components can more easily identify which components can be successfully moved to public and private clouds. A *service oriented architecture (SOA)* is a software architecture for building business applications that implement business processes or services through a set of loosely coupled, black-box components orchestrated to deliver a well defined level of service. This approach lets companies leverage existing assets and create new business services that are consistent, controlled, more easily changed, and more easily managed. SOA is a business approach to designing efficient IT systems that support reuse and give the businesses the flexibility to react quickly to opportunities and threats.

Characterizing SOA :

The principal characteristics of SOA are described in more detail here:

- **SOA is a black-box component architecture.** The *black box* lets you reuse existing business applications; it simply adds a fairly simple adapter to them. You don’t need to know every detail of what’s inside each component; SOA hides the complexity whenever possible.

- **SOA components are loosely coupled.** Software components are *loosely coupled* if they’re designed to interact in a standardized way that minimizes dependencies. One loosely coupled component passes data to another component and makes a request; the second component carries out the request and, if necessary, passes data back to the first. Each component offers a small range of simple services to other components. A set of loosely coupled components does the same work that software components in tightly structured applications used to do, but with loose coupling you can combine and recombine the components in a bunch of ways. This makes a world of difference in the ability to make changes easily, accurately, and quickly.

- **SOA components are orchestrated to link through business processes to deliver a well-defined level of service.** SOA creates a simple arrangement of components that, together, deliver a very complex business service. Simultaneously, SOA must provide acceptable service levels. To that end, the components ensure a dependable service level. Service level is tied directly to the best practices of conducting business, commonly referred to as *business process management (BPM)* — BPM focuses on effective design of business process and SOA allows IT to align with business processes.

Open Source Software in Data Centers:

Refer Page number 75 to 100 of ***Cloud Computing: Implementation Management and Security****, John W. Rittinghouse and James F. Ransome*