Service Oriented Architecture

Service-oriented architecture (SOA) is a software development model that allows services to communicate across different platforms and languages to form applications.

Service-oriented architecture allows various services to communicate using a loose coupling system to either pass data or coordinate an activity.

In this architecture, services are provided to form applications, through a network call over the internet. It uses common communication standards to speed up and streamline the service integrations in applications.

Each service in SOA is a complete business function in itself. The services are published in such a way that it makes it easy for the developers to assemble their apps using those services.

A basic SOA architecture is composed of a service provider, service and an optical service directory. Application-to-application messaging is used in the information exchange.

The similarity between this model and that of straight web services is very visible, with WSDL being the invocation contract stored in a service directory where it can be queried and fetched via UDDI.

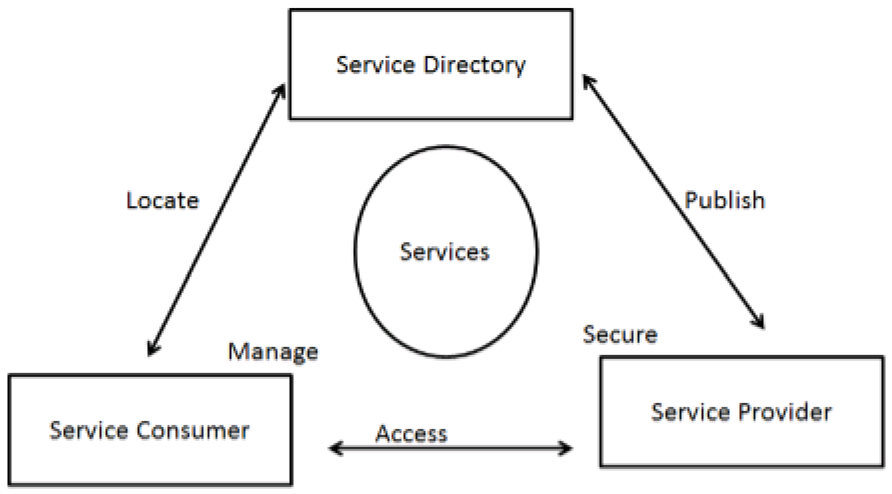

Fig: SOA ARCHITECTURE

In this model, the basic scenario is as follows: First the service provider creates a service and decides to expose it and publishing is done by posting the service information on the service directory.

On other side, a service requester, in need of certain, searches the service directory for one that meets the necessary criteria.

Upon finding one and using the information available on the service directory, the service requester is able to directly contact the service provider in the correct way to fulfill the business need.

Here are some definitions of terms used in this section:

Service provider: provider of services whose invocation contract and location are published.

Service consumer: consumer of services matching his or her business need found in a service directory

Service directory: Directory for publishing and listing available service for consumers

Characteristics of SOA are as follows:

- Provides interoperability between the services.
- Provides methods for service encapsulation, service discovery, service composition,
- service reusability and service integration.
- Facilitates QoS (Quality of Services) through service contract based on Service Level
- Agreement (SLA).
- Provides loosely couples services.
- Provides location transparency with better scalability and availability.
- Ease of maintenance with reduced cost of application development and deployment.

Disadvantages of SOA:

- High overhead: A validation of input parameters of services is done whenever services interact this decreases performance as it increases load and response time.
- High investment: A huge initial investment is required for SOA.

- Complex service management: When services interact they exchange messages to tasks. the number of messages may go in millions. It becomes a cumbersome task to handle a large number of messages.

Resource-Oriented Architecture (ROA) paradigm is built on the notion of a resource. A resource is a self-contained, identifiable entity having a state that can be assigned a uniform resource locator (URI).

While a service represents the execution of a requested action, a resource represents a distributed component that is manageable through a consistent, standardized interface.

One of the characteristics of resource-oriented architecture is that they are transport-independent. So, there must be specific mechanisms for exposing resource-oriented services to the external world.

When a consumer requests a Uniform Resource Locator (URL) and specifies an access method (e.g., GET, PUT, POST, & DELETE), the URL gets transformed into a relative internal URI. The access method is translated into action.

Resource-Oriented Architecture is modeled around just four concepts:

- Resources
- URIs
- Representations
- Links and Connectedness

Here are the four properties of Resource Oriented Architecture:

- Addressability
- Statelessness
- Connectedness
- A uniform interface