SOA - Introduction:

Service-Oriented Architecture (SOA) is a new way of thinking about enterprise IT Architecture. SOA is about associating business process with IT. SOA represents a natural evolution of proven software architectural principles or design patterns commonly implemented in object-oriented systems.

In short SOA is the concept that defines an architecture which gives a conduit to deliver business processes as services and increase the growth for business applications.

Here we will emphasize on functional aspects of architecture and advantages of SOA.

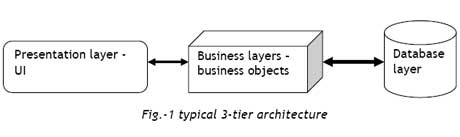
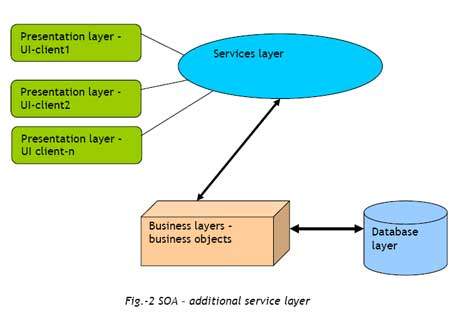
Business applications are developed in an organization as a solution to a specific business process. SOA promotes the idea that IT applications are service providers and consumers acting together to support the overall business goals of an organization. This new approach to enterprise architecture is a loosely-coupled solution in which software components are very independently.

Service Oriented Architecture and Design:

Service-oriented architecture is collection of many services in the network. These services communicate with each other and the communications involves data exchange & even service coordination. Earlier SOA was based on the DCOM or Object Request Brokers (ORBs). Nowadays SOA is based on the Web Services

In a service-oriented design, services should be course-grained. Course-grained services are modeled after and align to business processes. Course-grained services are business functions that require very little communication, and each service is designed to do a relatively large amount of work. Consumers of course-grained services are required to make minimal service calls to accomplish the desired functionality.

We can compare SOA with typical 3-tier architecture precisely in following diagrams

# 3-tier:

In 3-tier architecture we have 3 tiers/layers for segregation of work; if we consider a web application as an example 3 tiers would be,

Presentation: web UI i.e. web pages which will be available in internet or client end.

Business: This consists of classes, business logic, objects interacting with database and front end i.e. UI; the code written in classes for various calculations and generating results for requests from upper presentation layer.

Database Layer: will hold actual database and data access objects to which business layer can refer.

# SOA:

In a service-oriented architecture, clients consume services, rather than invoking discreet method calls directly.

In a service-oriented model, we introduce a further "layer of indirection". This alleviates some of the pain associated with traversing complex object models. The services layer above business layer.

SO the service layer would provide collection of different services which can be referenced by clients or consumers. Services are responsible for orchestrating calls to discreet business objects, managing the responses, and acting accordingly. Service methods may invoke and manage several business objects.

## SOA deployment and Enterprise Service Bus (ESB):

The deployment of a SOA requires the conversion of existing systems into services. The tasks involved in achieving this can be repeated for each system and a common set of components may be needed to provide additional functionality (such as security and auditing). An Enterprise Service Bus (ESB) is a collection of servers and components that provide these functions and a set of tools to assist in the conversion of existing systems to services.

## Service-Oriented Architecture and Web Services:

Web Services are not inherently service-oriented they represent a solution to implementing a SOA strategy. Web Services provide a necessary standards-based implementation of SOA where in services are described using WSDL and accessed via SOAP. So that interoperability across platform, organizational, and network boundaries is feasible.

Web Services are logic units that allow programs written in different programming languages and on different platforms to communicate through standard Internet protocols.

Web Services currently represent the best technology for implementing SOA.

# Microsoft.NET and Web Services:

Although Web Services can be created on any platform, Microsoft's .NET framework is heavily geared towards the creation of Web Services. It provides a number of tools and features through which a Web Service can be quickly and easily built. The .NET framework offers a high degree of native XML support; it is well suited to building Web Services. The framework will also automatically generate the WSDL required.

# Conclusion:

SOA is an architecture which evolved from the distributed and the components architectures and has become the new one to solve complex problems around the enterprise environment.

How Service-Oriented Architectures Can Benefit a Business

The SOA could hold true for ecommerce and service industry that includes scheduling, registration, and credit card processing.s

Key advantages of SOA:

* reusability
* platform independence
* incremental development, deployment, and maintenance.