* **SOA Definition and Characteristics**

**Ans : -**

A service-oriented architecture 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. Some means of connecting services to each other is needed.

Service-oriented architectures are not a new thing. The first service-oriented architecture for many people in the past was with the use DCOM or Object Request Brokers (ORBs) based on the CORBA specification.

**Service-oriented architectures have the following key characteristics:**

* SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language ([WSDL](http://www.javaworld.com/#resources)) is the standard used to describe the services.
* SOA services communicate with messages formally defined via XML Schema (also called [XSD](http://www.javaworld.com/#resources)). Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider. Messages between services can be viewed as key business documents processed in an enterprise.
* SOA services are maintained in the enterprise by a registry that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration ([UDDI](http://www.javaworld.com/#resources)) is the standard used for service registry.
* Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services.

### Features and Benefits of SOA

SOA is a software architecture that determines the features that make up an application and should be made available as services that communicate with each other through messages. That way, applications can be developed into small parties, facilitating management of development teams. But SOA is more than that—it is a robust architecture, focused on the integration of systems, whose main benefits are discussed below:

* Integration between different platforms: With exchanging XML messages between services is possible to integrate different platforms, such as Java and PHP. The XML language is widely used for data exchange, XML is basically a text file with the data and its meta-information tags.
* Low coupling: As services are developed to meet a specific demand, the level of cohesion is relatively high, so a program developed for SOA is composed of several parts that are well defined to be accounted for only as functions that operate independently.
* Code reuse: The low coupling and high cohesion allow each service to be used in several different systems without rework or large efforts of the teams involved.
* Ease of construction and maintenance: Save time by reusing services and the possibility of change within a point of the system; in case of future maintenance or errors, correction dramatically decreases the time unavailability in the system.
* Ease distribution of functionalities: It's necessary to disclose the service descriptor, and features will be available to any developer who needs to use them.