



A

Introducing SOA

Service Oriented Architecture (SOA) defines how different components of a software application interact to execute the business logic of an enterprise application. In other words, SOA is a software design pattern that defines how loosely-integrated software application components interact to execute the business logic of the entire application. In SOA, the independent software application components involved in executing the business logic are termed as software services, or simply services. Services may be defined as independent and self-sustained units of a software application that implement specific functionalities to execute the business logic. These functionalities are generally a part of the overall business logic of the software application. For example, consider a software application of a bank. The bank can provide specific functionalities, such as submitting an online registration form for opening a new bank account and viewing the bank statement. These functionalities can be implemented using independent services. This means that you can have independent services to submit an online form and view the bank statement. Though these services work independently, they are part of the banking software application. SOA, as a design pattern, defines how these individual and independent services are suitable for the banking software application.

SOA primarily focuses on binding large and independent services, which normally interact through interfaces. These services are known as Web services, which are interoperable software applications that work on the client/server model. As the name suggests, Web services can be accessed over a network, such as the Internet, through standard communication protocols, such as Hyper Text Transfer Protocol (HTTP). A Web service performs the standard task of an application based on the client/server model; that means, receiving requests from a client, processing the requests, and providing the requested resources or responses. A simple eCommerce Web site, such as eBay, is an example of a Web service.

In this appendix, you learn about the architecture of SOA. You also learn about the SOA environment and its layers.

Architecture of SOA

SOA is a technology that makes possible to link a variety of resources on a network. It enables you to develop the applications that are built by combining loosely integrated (coupled) and interoperable Web services. Figure A.1 shows the architecture of SOA:

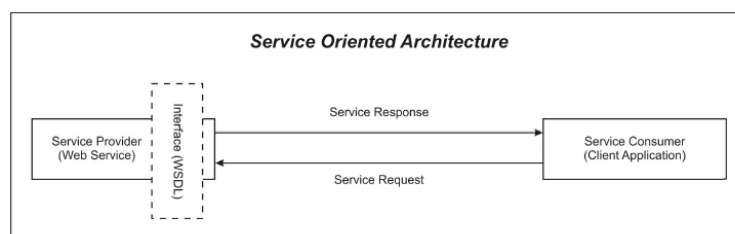


Figure A.1: Displaying the Architecture of SOA

As shown in Figure A.1, the software application implementing SOA works on the client/server model. This type of application generally contains a Web service, which is hosted on the server side and acts as the service provider. Service requests are sent to the service provider, which processes and returns a response message or requested resources back to the client application, also known as the service consumer. For example, consider a simple Web service used to convert currencies. The Web service contains the logic to accept input (in the form of a currency) and convert it into the required format (in terms of the other currency). This Web service, hosted on the server side, is the service provider. A Web browser is used to access the currency converter Web service, and this Web browser acts as the service consumer. When the Web browser (or service consumer) accesses the currency converter Web service (service provider) to process a conversion request, the process is termed as consuming the Web service.

SOA-based applications are platform independent, implying service providers can implement Web services in different platforms and languages, such as .NET or J2EE, and a service consumer or client application can use the service on a different platform or language. Enterprises can also plug in new services or upgrade existing services to implement new business requirements. The key characteristics of SOA are as follows:

- ❑ Interfaces of SOA-based applications are described using a platform-independent Extensible Markup Language (XML) document, known as a Web Services Description Language (WSDL). WSDL is an XML-based language used to describe Web services. The WSDL interface hides the implementation of language specific service; therefore, Web services maintain the interoperability feature.
- ❑ SOA-based applications use XML schemas to define the structure, content, and semantics of messages (XML documents), which are shared among services.
- ❑ SOA-based enterprise applications are registered in a service registry so that client applications can look up and invoke them. A service registry is defined by using Universal Description, Discovery and Integration (UDDI).
- ❑ Each SOA service is associated with a quality of service (QoS). Some elements of a QoS are security requirements (such as authentication and authorization), reliable messaging, and policies regarding the user who can invoke services.

Let's now explore the SOA environment in detail.

Describing the SOA Environment

In the previous section, you have learned about the architecture of SOA, which had one service provider and one service consumer. However, in real-life situations, SOA-based applications might span across numerous service providers and consumers. At times, big enterprise applications might also be based on SOA. These enterprises must conform to the SOA environment to run and manage SOA-based applications. The SOA environment specifies the required standards and runtime containers needed to implement SOA-based applications. Figure A.2 shows of the environment of SOA:

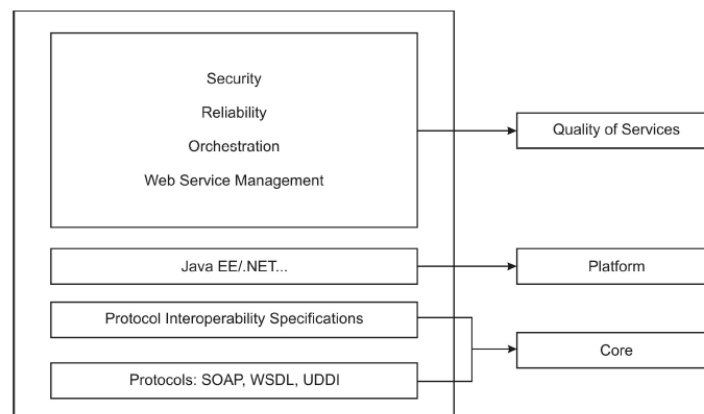


Figure A.2: Displaying the SOA Environment

In Figure A.2, you see that the environment of SOA comprises of the following layers:

- ❑ Core
- ❑ Platform
- ❑ Quality of Services

Let's now explore these layers in detail.

The Core Layer

The core layer of the SOA environment includes the following components:

- ❑ **Protocol** – Refers to the standards that are followed by Web services for communication. SOA rests on the following protocols:
 - **Simple Object Access Protocol (SOAP)** – Acts as a transport layer protocol used to transmit messages between service clients and service providers.
 - **WSDL** – Provides a format to describe the Web service interfaces. A WSDL file contains data type and message definitions, and defines how the Web services need to be bound to specific network addresses. To interact with a SOA-based Web service, service consumers need to obtain WSDL of the Web service and then call the service by using SOAP.
 - **UDDI** – Defines a way to publish and discover information about Web services. UDDI is used by service providers to register services on a Web server. Service consumers also use UDDI to search Web services in the UDDI registry.
- ❑ **Protocol Interoperability Specification** – Refers to a specification that defines the standards for SOAP, WSDL, and UDDI protocols to ensure interoperability of Web services. It is known as WS-I Basic Profile, which contains implementation guidelines for basic Web Services standards. Nowadays, the Web Services industry provides new specifications, such as WS-Security, WS-ReliableMessaging (WS-RM), and WS-AtomicTransactions (WS-AT). These specifications provide advanced functionalities such as security, reliability, and transactions that are not provided by WS-I Basic Profile. As a result, these specifications are popular as WS-* (pronounced WS-Star) specifications.

The Platform Layer

The platform layer contains the different development platforms, such as Java EE and .NET, used to develop SOA-based applications. SOA enables you to create services in .NET by using C# language and in JavaEE by using Java language. These services can be consumed by a common client application. SOA Web services are interoperable as applications running on either platform can also consume services running on the other.

The Quality of Services Layer

The Quality of Services layer of the SOA environment caters to requirements of enterprise applications, such as security, reliability, and transaction management. Organizations, such as World Wide Web 3 Consortium (W3C) and Organization for the Advancement of Structured Information Standards (OASIS), have developed many specifications related to the quality of Web services, such as:

- ❑ Security
- ❑ Reliability
- ❑ Policy
- ❑ Orchestration
- ❑ Web Service Distributed Management (WSDM)

Let's discuss these in detail.

Security

The Web Service Security specification defines processes and methods to secure messages in various ways, such as by authenticating messages based on certain user credentials or using Security Assertion Markup Language (SAML) to secure Web service messages.

Reliability

Reliability means delivering a message to the recipient with an acknowledgement back to the sender and avoiding delivery of duplicate messages. The WS-Reliability and WS-ReliableMessaging standards handle reliability of Web service messages.

Policy

The WS-Policy standard defines a set of rules that an application uses to process a WS-Policy message. These set of rules are called policy standards or policy rules, which must be followed to exchange Web service messages between service providers and service consumers. For example, a policy rule can encrypt a request to a Web service or declare the maximum acceptable size of a message by a Web service.

Orchestration

Web Service Orchestration (WSO), or simply orchestration, defines the interaction, automated arrangement, and management of different components of a SOA-based application. In specific, WSO defines the deployment structure of Web service components of a SOA-based application to automatically process the required business logic. Consequently, WSO also ensures that a SOA-based application is scalable, implying new components can be added or existing components can be modified with minimum effort as and when a business need arises. In other words, WSO can be considered as an extension of business process management, which defines the interaction of individual components of a software application to process the business logic. WSO is implemented by using different programming languages, such as Business Process Execution Language for Web Services (BPEL4WS) and Web Service Choreography Interface (WSCI).

Web Service Management

The status of Web services needs to be managed and monitored when the number of Web services increases in a SOA-based enterprise application. For example, you can manage and monitor the status of a Web service by verifying the various parameters, such as:

- ❑ The state of the Web service, i.e. whether or not the Web service is processing client requests
- ❑ The number of requests processed
- ❑ The number of requests that could not be processed due to some errors
- ❑ The number of requests that have been responded
- ❑ The number of requests that have a limited time period

You need to monitor the status of Web services according to the Web Services for Distributed Management (WSDM) standard, so that the WSDM-complaint management infrastructure of Web services can manage services running in a heterogeneous environment. The Web Service (WS)-Coordination and WS-Transaction specifications manage the interaction between service providers and consumers, and handle transactions involving numerous Web services. You can integrate third-party Web service monitoring tools, such as Actional SOAPstation developed by Progress Actional, with Web services in a SOA-based application to manage and monitor the status of Web services.

With this, we have come to the end of this appendix.