**SOA**

Service-oriented architecture (SOA) is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications. An application's business logic or individual functions are modularized and presented as services for consumer/client applications. What's key to these services is their loosely coupled nature; i.e., the service interface is independent of the implementation. Application developers or system integrators can build applications by composing one or more services without knowing the services' underlying implementations. For example, a service can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.

Service-oriented architectures have the following key characteristics:

* SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WSDL) is the standard used to describe the services.
* SOA services communicate with messages formally defined via XML Schema (also called XSD). 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) 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.

### Why SOA?

The reality in IT enterprises is that infrastructure is heterogeneous across operating systems, applications, system software, and application infrastructure. Some existing applications are used to run current business processes, so starting from scratch to build new infrastructure isn't an option. Enterprises should quickly respond to business changes with agility; leverage existing investments in applications and application infrastructure to address newer business requirements; support new channels of interactions with customers, partners, and suppliers; and feature an architecture that supports organic business. SOA with its loosely coupled nature allows enterprises to plug in new services or upgrade existing services in a granular fashion to address the new business requirements, provides the option to make the services consumable across different channels, and exposes the existing enterprise and legacy applications as services, thereby safeguarding existing IT infrastructure investments.

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. For more on DCOM and CORBA

**Web service:**

"Web service" is “a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a specific format (specifically Web Services Description Language, known by the acronym WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

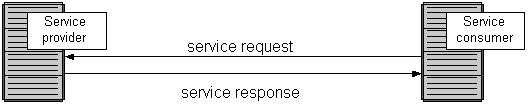
**Services:**

A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services.

## Connections:

The technology of Web services is the most likely connection technology of service-oriented architectures. Web services essentially use XML to create a robust connection.

The following figure illustrates a basic service-oriented architecture. It shows a service consumer at the right sending a service request message to a service provider at the left. The service provider returns a response message to the service consumer. The request and subsequent response connections are defined in some way that is understandable to both the service consumer and service provider. A service provider can also be a service consumer.

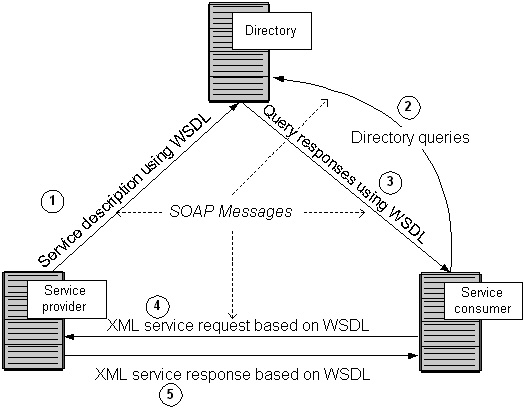


**Web Services explained:**

**Using the Web Services Description Language (WSDL)**

The Web Services Description Language (WSDL) forms the basis for Web Services. The following figure illustrates the use of WSDL. At the left is a service provider. At the right is a service consumer. The steps involved in providing and consuming a service are:

1. A service provider describes its service using WSDL. This definition is published to a directory of services. The directory could use Universal Description, Discovery, and Integration (UDDI). Other forms of directories can also be used.
2. A service consumer issues one or more queries to the directory to locate a service and determine how to communicate with that service.
3. Part of the WSDL provided by the service provider is passed to the service consumer. This tells the service consumer what the requests and responses are for the service provider.
4. The service consumer uses the WSDL to send a request to the service provider.
5. The service provider provides the expected response to the service consumer.



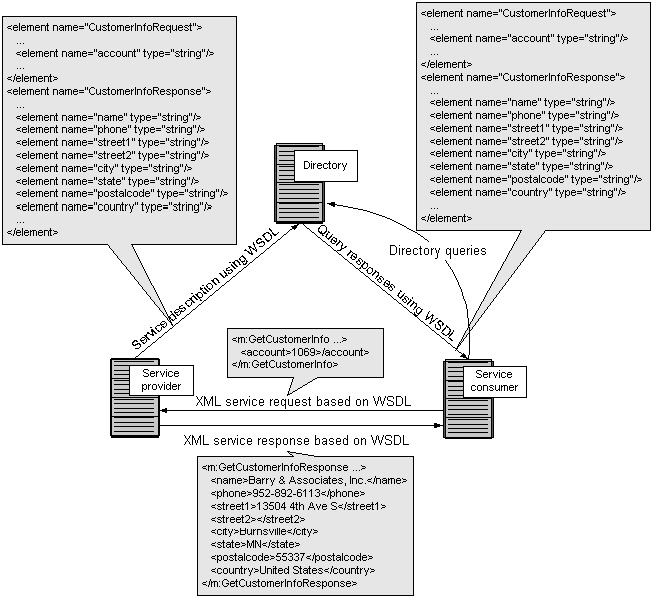
## Using Universal Description, Discovery, and Integration (UDDI)

The directory shown in the above figure could be a UDDI registry. The UDDI registry is intended to eventually serve as a means of "discovering" Web Services described using WSDL . The idea is that the UDDI registry can be searched in various ways to obtain contact information and the Web Services available for various organizations. How much "discovery" will be used in the early days of Web Services is open to discussion. Nevertheless, even without the discovery portion, the UDDI registry is a way to keep up-to-date on the Web Services your organization currently uses.

## Using SOAP

All the messages shown in the above figure are sent using SOAP. (SOAP at one time stood for Simple Object Access Protocol. Now, the letters in the acronym have no particular meaning .) SOAP essentially provides the envelope for sending the Web Services messages. SOAP generally uses HTTP , but other means of connection may be used. HTTP is the familiar connection we all use for the Internet. In fact, it is the pervasiveness of HTTP connections that will help drive the adoption of Web Services.

The next figure provides more detail on the messages sent using Web Services. At the left of the figure is a fragment of the WSDL sent to the directory. It shows a CustomerInfoRequest that requires the customer's account to object information. Also shown is the CustomerInfoResponse that provides a series of items on customer including name, phone, and address items.



At the right of this figure is a fragment of the WSDL being sent to the service consumer. This is the same fragment sent to the directory by the service provider. The service consumer uses this WSDL to create the service request shown above the arrow connecting the service consumer to the service provider. Upon receiving the request, the service provider returns a message using the format described in the original WSDL. That message appears at the bottom of the figure.

## Using XML with WSDL

WSDL uses XML to define messages. XML has a tagged message format. This is shown in the above figure. The tag <city> has the value of Burnsville. And </city> is the ending tag indicating the end of the value of city. Both the service provider and service consumer use these tags. In fact, the service provider could send the data shown at the bottom of this figure in any order. The service consumer uses the tags and not the order of the data to get the data values.

## Simplified Web Services notation

For other figures on this site, a simplified notation will be used for Web Services. This is shown below. In the simplified notation, the directory is implicit in the wide rectangle labeled "Web Services" at the top of this figure. You could think of Web Services much like the bus in a PC in which you plug various circuit boards. Other middleware solutions appear similar and use the same "bus" concept.

Another important concept in service-oriented architectures is that any service provider could also be a service consumer. This is why the figure below shows only services at the bottom of the figure under the Web Services bus rather than a "service provider" and a "service consumer."

