# Service-Oriented Architecture (SOA) Definition

# Web Services Definition

The term "Web Services" can be confusing. It is, unfortunately, often used in many different ways. Compounding this confusion is term "services" that has a different meaning than the term "Web Services." On this site, the term*Web Services* refers to the technologies that allow for making connections. *Services* are what you connect together using Web Services. A service is the endpoint of a connection. Also, a service has some type of underlying computer system that supports the connection offered. The combination of services—internal and external to an organization—make up a *service-oriented architecture*.

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.

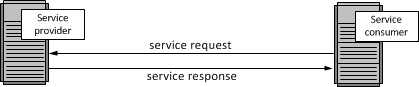
Service-oriented architectures are not a new thing. The first service-oriented architecture for many people in the past was with the use DCOM or Object Request Brokers (ORBs) based on the CORBA specification. For more on DCOM and CORBA, see [Prior Service-Oriented Architectures.](http://www.service-architecture.com/articles/web-services/prior_service-oriented_architecture_specifications.html)

## Services

If a service-oriented architecture is to be effective, we need a clear understanding of the term service. A service is a function that is well-defined, self-contained, and does not depend on the context or state of other services. See[Service.](http://www.service-architecture.com/articles/web-services/service.html)

## Connections

The technology of [Web Services](http://www.service-architecture.com/articles/web-services/web_services_definition.html) is the most likely connection technology of service-oriented architectures. 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. How those connections are defined is explained in [Web Services Explained.](http://www.service-architecture.com/articles/web-services/web_services_explained.html) A service provider can also be a service consumer.



Web Services Explained

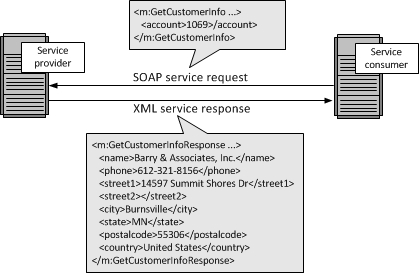
First, Web Services using SOAP, REST, and JSON are discussed. This is followed by a history of Web Services covering the Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI).

Web Services Specifications

Three specifications for Web Services are illustrated in this section: SOAP, REST, and JSON.

SOAP

SOAP was originally part of the specification that included the Web Services Description Language (WSDL) and Universal Description, Discovery, and Integration (UDDI). It is used now without WSDL and UDDI. Instead of the discovery process described in the History of the Web Services Specification section below, SOAP messages are hard-coded or genereated without the use of a repository. The interaction is illustrated in the figure below. More on[SOAP](http://www.service-architecture.com/articles/web-services/soap.html).



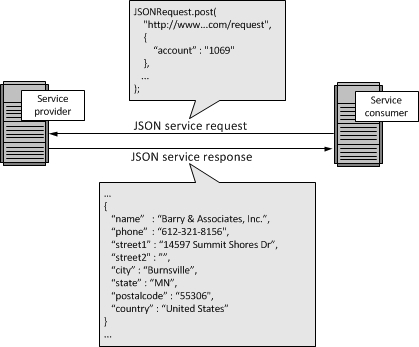
### Representation State Transfer (REST)

Representation State Transfer (REST) appeals to developers because it has a simpler style that makes it easier to use than SOAP. It also less verbose so that less volume is sent when communicating. The interaction is illustrated in the figure below. More on [REST](http://www.service-architecture.com/articles/web-services/representational_state_transfer_rest.html).



### JavaScript Object Notation (JSON)

While both SOAP and REST use XML for interchange, JavaScript Object Notation (JSON) uses a subset of JavaScript. This is illustrated in the figure below. More on [JSON](http://www.service-architecture.com/articles/web-services/javascript_object_notation_json.html).



### When to Use SOAP, REST, JSON or Other Options

There really is no "best" option for Web Services. Generally, you will use whatever your service provider supports. If you use multiple service providers, it is easily possible that you will be using all three Web Services specifications: SOAP, REST, and JSON.

## History of the Web Services Specification

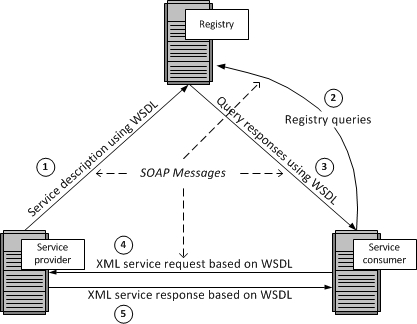
Web Services Description Language (WSDL); Universal Description and Discovery (UDDI); and SOAP formed the original Web Services specification. This section provides a history.

### Web Services Description Language (WSDL)

The Web Services Description Language (WSDL) forms the basis for the original Web Services specification. 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 repository of services. The repository could use Universal Description, Discovery, and Integration (UDDI). Other forms of directories could also be used.
2. A service consumer issues one or more queries to the repository 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.

More on [Web Services Description Language.](http://www.service-architecture.com/articles/web-services/web_services_description_language_wsdl.html)



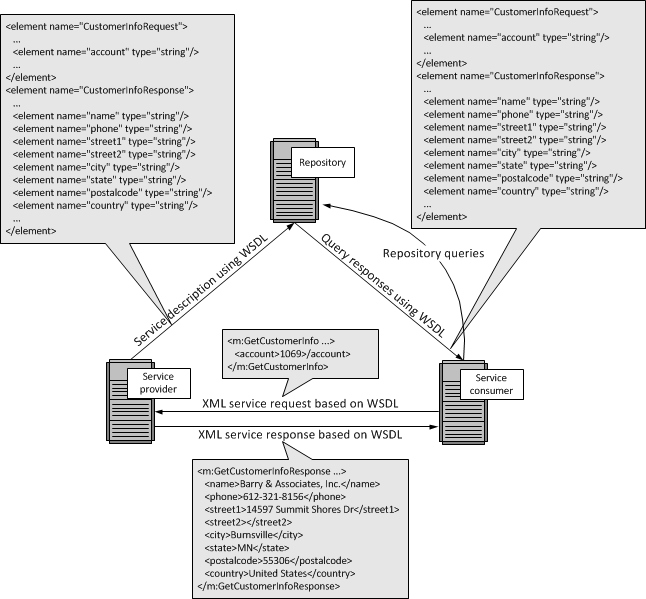
### Universal Description, Discovery, and Integration (UDDI)

The repository shown in the above figure could be a UDDI registry. The UDDI registry was 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" was ever used 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. It can be used at design time and with governance. An alternative to UDDI is the [ebXML Registry.](http://www.service-architecture.com/articles/web-services/ebxml_registry.html" \t "_blank) More on [Universal Description, Discovery, and Integration.](http://www.service-architecture.com/articles/web-services/universal_description_discovery_and_integration_uddi.html)

### 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. More on SOAP and [Messaging.](http://www.service-architecture.com/articles/web-services/messaging.html)

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 repository. 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 repository 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.

XML is used to define messages. XML has a tagged message format. You can see this in the SOAP and REST examples in the first section and in the figure above. In each of the examples, 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. More on the use of XML tags and a comparison of XML to using fixed record formats can be found in chapter 3 of [*Web Services, Service-Oriented Architectures, and Cloud Computing: The Savvy Manager's Guide.*](http://www.barryandassociates.com/managers-guide.html) Also see [XML Processing](http://www.service-architecture.com/articles/xml/xml_background.html) for an overview of the XML processing in a service.