

SOAP vs REST

SOAP is a protocol specification for exchanging structured information in the implementation of Web Services. It uses XML for the message format. It is independent of the transport protocol (could be HTTP, FTP, TCP, UDP, or named pipes). SOAP based services strictly define the format of messages passed back and forth. A SOAP message contains the data, the action to perform on it, the headers, and the error details in case of failure. Security is provided by WS-Security standards and is end-to-end. It supports identity through intermediaries, not just point to point (SSL).

Representational State Transfer (REST) is an architecture style for designing networked applications. REST recognises everything a resource (e.g. User, Course, etc.) and each resource implements a standard uniform interface (typically HTTP interface), resources have name and addresses (URIs), each resource has one or more representation (like JSON or XML) and resource representations move across the network usually over HTTP.

RESTful web APIs (or RESTful web service) is a web API implemented using HTTP and principles of REST. RESTful API separates user interface concerns from data storage concerns. It improves portability of interface across multiple platforms and simplifies server components by making them stateless. Each request from client contains all the state information and server does not hold client context in the session.

One of the major benefits of SOAP is that you have a WSDL service description, like a contract between the two parties. You can pretty much discover the service automatically and generate a useable client proxy from that service description (generate the service calls, the necessary data types for the methods and so forth). With REST you'll have to implement such contract yourself in your code.

Note that with version 2.0, WSDL supports all HTTP verbs and can be used to document RESTful services as well, but there is a less verbose alternative in WADL (Web Application Description Language) for that purpose.

With RESTful services, message security is provided by the transport protocol (HTTPS), and is point-to-point only. It doesn't have a standard messaging system and expects clients to deal with communication failures by retrying. SOAP has successful/retry logic built in and provides end-to-end reliability even through SOAP intermediaries.

One of the major benefits of RESTful API is that it is flexible for data representation, for example you could serialize your data in either XML or JSON format. RESTful APIs are cleaner or easier to understand because they add an element of using standardised URIs and gives importance to HTTP verb used (i.e. GET, POST, PUT and DELETE).

RESTful services are also lightweight, that is they don't have a lot of extra XML markup. To invoke RESTful API all you need is a browser or HTTP stack and pretty much every device or machine connected to a network has that. With web services that are generally SOAP-based, the request and response are hidden. SOAP requests must be interpreted as they are received at the server to determine the operation to perform and the arguments required to perform that operation. They are generally passed through as a parameter, which is essentially a function/method call. SOAP does not currently provide a mechanism for caching results, where as REST can (through the standard HTTP caching).

Essentially, the Traditional SOA way of doing things and the RESTful way of doing things would both yield similar results. However you might find that the Traditional SOA would allow you to do more powerful things (and as a result is more complex) and the RESTful oriented architecture would allow you to easily do simple things (and this being much easier to implement) REST becomes an architecture style applicable to systems that would want operate with a Resource view. SOA, on the flip-side, has a different view of the web. It uses

the web as a application infrastructure between service providers and consumers otherwise known as the Service view.

In conclusion, we can say that web services standards provide an open standards based communication framework that operates within the purview of W3C guidelines. Web Services provide a platform/technology independent communication methodology while SOA is an overall IT strategy framework that aims to provide business agility. Finally, which ever architecture you choose make sure its easy for developers to access it, and well documented.

[Prev](#)

SOAP Web Services

[Up](#)[Home](#)[Next](#)

Chapter 7. References