What is the difference between SOAP and REST?

SOAP Overview

SOAP (Simple Object Access Protocol) is the foundation layer of a web services protocol stack, providing a basic messaging framework upon which web services can be built. This XML-based protocol consists of three parts: 1) an envelope, which defines what is in the message and how to process it; 2) a set of encoding rules for expressing instances of application-defined data types; and 3) a convention for representing procedure calls and responses. For example, a SOAP message could be sent to a web site that has web services enabled, such as a real-estate price database. The message would contain the information required for a search. The site would return an XML-formatted document with the resulting data, like prices, location, and features. This standardized machine-parsable data can then be integrated directly into a third-party web site or application.

The XML messaging that SOAP employs works better over the Internet than SOAP's predecessors. However, the XML used to make requests and receive responses in SOAP can become extremely complex. In some programming languages, like .NET languages, the developer may never see the XML. The Web Services Description Language (WSDL) provides a definition of how the Web service works, so that when a reference is created for the WSDL, the IDE can completely automate the process.

A Quick Overview of REST

REST (REpresentational State Transfer) evokes an image of how a well-designed Web application behaves: The user progresses through the application by selecting links, or state transitions. This causes the rendering of the next page of a network of Web pages, using a virtual state-machine paradigm.

REST was initially described in the context of HTTP, but it is not limited to that protocol. REST is a light weight alternative to SOAP; instead of using XML, REST relies on a simple URL in many cases. Most Web services using REST rely exclusively on the URL approach - the familiar HTTP 1.1 verbs (GET, POST, PUT, and DELETE) to perform tasks. REST is not required to use XML to provide the response. REST-based Web services can output the data in Command Separated Value (CSV) format, in JavaScript Object Notation (JSON), or Really Simple Syndication (RSS). All of these formats are easily parsed within several application languages.

Deciding Between SOAP and REST

In most enterprises, Web service models have already been selected, so it is a good idea to become familiar with both SOAP and REST. However, if you need to select a protocol, you may want to consider the following factors.

Advantages of SOAP

- Language, platform, and transport independent (REST requires use of HTTP)
- Works well in distributed enterprise environments (REST assumes direct point-to-point communication)
- Standardized
- Provides significant pre-build extensibility in the form of the WS* standards
- Built-in error handling
- Automation when used with certain language products
- Sometimes preferred for services within an enterprise, especially for interoperability between servers that cannot directly communicate

Disadvantages of SOAP

- Verbose XML format
- Can be considerably slower than competing middleware technologies
- Many developers found SOAP cumbersome and hard to use
- Requires the XML structure to be re-created every time

Advantages of REST

- REST is easier to use and is more flexible than SOAP
- No expensive tools are required for Web service interaction
- Smaller learning curve
- Efficient, small message formats
- Fast (no extensive processing required)
- Good for creating useful web services quickly
- Does not require a strict API definition

Disadvantages of REST

- REST is considered an architectural approach
- May decrease network performance by increasing repetitive data
- Data cannot be left on the server in a shared context
- Servers and clients implementing/using REST are vulnerable to the same threats as any HTTP/Web application

Both SOAP and REST are protocols: a set of rules for requesting information from a server using a specific technique. A number of factors should be considered when selecting a protocol for a new Web client: server interoperability, client language compatibility, prototyping and development speed, and network performance. Engineers should consider these and other architectural requirements when specifying new Web services that will require SOAP or REST interfaces.

Sources

http://en.wikipedia.org/wiki/SOAP

http://en.wikipedia.org/wiki/REST

http://blog.smartbear.com/apis/understanding-soap-and-rest-basics/

http://www.nsa.gov/ia/_files/support/guidelines_implementation_rest.pdf