SOAP vs REST

Overview

This document will provide a comparison of SOAP and REST. It will also highlight which scenarios are best suited to each.

What is SOAP?

SOAP (Simple Object Access Protocol) transports XML documents over multiple different internet protocols. These include HTTP, SMTP & FTP. SOAP only supports data in XML format. Web Services use this protocol to communicate.

Key Functionality of SOAP

SOAP provides Web Services Security as well as SSL
SOAP has built in success/retry logic which provides end to end reliability.
SOAP uses service interfaces to expose business logic.
SOAP has built in state management

When to use SOAP

1. SOAP has tighter security than REST, communications between client and server are very rigid, any change from either side deems the contract null. SOAP should be used with banking or apps that require reliable communication or ACID (Atomicity, Consistency, Isolation, Durability) compliance.
2. Asynchronous Processing – SOAP provides Web Service Reliable Messaging for guaranteed level of reliability.

Example uses of SOAP

Financial Services, Telecommunications Services and Payment Gateways

What is REST?

REST (Representational State Transfer) is an architectural style based on the following six constraints:

1. Uniform interface - REST is defined by four interface constraints. These are identification of resources; manipulation of resources through representations; hypermedia as the engine of the application and self-descriptive messages.
2. Stateless - The session state is contained entirely on the client, there's no state context stored on the server. All requests must be self-contained with all relevant data.

3. Cacheable - Cache requirement states that the response data must either be marked cacheable or non-cacheable. If the data is cacheable the client has been granted the right to reuse the data in later equivalent requests.
4. Client-Server - UI and data separation improves the portability of the UI and makes it multi-platform. Simplifying the server components improves the scalability of the system.
5. Layered System - Client is unaware if they are connected to the main server or an intermediate server. This allows for greater security measures.
6. Code on Demand (optional) - Servers can extend or customize the functionality of the client temporarily by executing code, e.g. An applet or JavaScript.

Key Functionality of REST

REST is resource based not action based
REST permits many different data formats
REST offers SSL security.
REST uses URI to expose business logic

When to use REST

1. REST lends itself well to caching situations. If data can be cached, then REST is a good fit due to its stateless nature.
2. Stateless operations e.g. Create, Read, Update Delete.
3. Limited Bandwidth and Resources – REST allows the use of any browser due to REST using standard GET, PUT, POST and DELETE. The return data can be in any format defined.
4. Mobile applications require speed and efficiency, JSON is easier to parse than XML using less CPU and memory. REST supports JSON and has a minimal overhead.

Example uses of REST

Web Chat Services, Mobile Services and Social Media Services

Conclusion

The main differences between SOAP and REST are as follows:

1. SOAP only supports XML Data whereas REST supports many data formats including JSON, XML, HTML or other predefined data formats.
2. REST is resource based, SOAP is action based.
3. SOAP offers more security – SSL & Web Service Security. REST only offers SSL.
4. REST is stateless, SOAP has built in state management..
5. REST uses URIs to expose business logic, SOAP uses Service Interfaces.

An over simplified analogy of SOAP vs REST that is often used when comparing them is to liken SOAP to an envelope and REST to a postcard, inferring that an envelope requires a lot more effort (extra overhead, more bandwidth required on both ends) to be mailed in comparison to a postcard (lightweight, ability to be cached and easy to update).

If your project involves payments, telecommunication services or financial services then SOAP is more suited but if your project involves social media, web chat services or mobile services then REST would be a better fit.