REST vs SOAP

Application Programing Interfaces (APIs) are integral to modern-day Internet communications. They facilitate interactions and transmission of data between applications without regard to platforms, programming languages, or other technologies.

APIs involved in web communications use the HTTP protocol and are known as web services. To the developer community, building or integrating applications with these APIs requires a knowledge of two primary types: SOAP (Simple Object Access Protocol) and REST (Representational State Transfer).

SOAP (Simple Object Access Protocol)

The SOAP protocol was one of the first API formats and was designed as a standardized way to send and receive messages and data between programs. SOAP APIs follow specifications that define the message format and how it is processed. In short, SOAP APIs use XML (extensible markup language) to format the message. The messages are made up of the envelope, header, body, and fault sections:

- Envelope: Defines the message as a SOAP API
- Header: An optional section that further defines the message and its requirements.
- Body: The content of the message that contains either requests (such as, methods and parameters to execute on the target resource) or response (the data returned by the request)
- Fault: Section that defines errors, if they occur during the transmission or execution of the message.

Example SOAP Request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Header>
  </soap:Header>
    <soap:Body>
    <m:GetRecord>
      <m:RecordId>123</m:RecordId>
    </m:GetRecord>
  </soap:Body>
</soap:Envelope>
```

REST (Representational State Transfer)

REST APIs evolved after the implementation of the SOAP protocols, and are defined by an architectural style, rather than a strict standard or specification. This style allows the use of many types of messaging formats, for example, XML, JSON, HTML, and so on. However, most REST APIs use JSON. A REST API uses a message format to access resources through a URL and common methods to access, create, or modify data (GET, POST, PUT, and DELETE). APIs that follow the REST style are often referred to as RESTful. At a high-level, the REST architectural style includes constraints to be considered fully RESTful including accessing client-server architectures, statelessness – that is, resources do not remember previous requests – and the ability to cache data for better performance.

Example REST Request:

```
http://apis_example_server.com/records?limit=3&format=json
```

Comparison

Both SOAP and REST APIs share similar characteristics. They both send messages to resources and receive responses; they both use the HTTP protocol; and they both can use XML in requests or responses.

However, the open architectural style of REST APIs provide more flexibility in design and is not tied to strict specifications like SOAP APIs. SOAP messages come with more overhead and bandwidth requirements, slowing processing and response time for calls, especially for mobile applications. For overall performance and flexibility REST APIs are now preferred by developers, although SOAP APIs continue to be implemented based on individual application requirements.