SOAP (Simple Object Access Protocol) is a web service protocol that uses XML and XML schemas. SOAP works only with XML. SOAP also only works with .NET Framework, not .NET Core. In contrast with REST (Representational State Transfer), all SOAP messages are sent over HTTP. A SOAP web service can be either stateful or stateless. A SOAP message is comprised of an Envelope, which contains two sections, the Header, and the Body. The header specifies the routing for the data, while the body contains the actual message data. A SOAP message must include an Envelope, and a Body. While the header is optional, if it is included, it must be the first child of the Envelope. While multiple headers are allowed in a single SOAP message, only one Envelope and one Body may be present. When creating a SOAP web service, .NET Framework uses WCF(Windows Communication Foundation).

REST is also an architectural design for web services. RESTful web services, while generally based on HTTP, are not bound to HTTP in the specifications. RESTful web services are also not bound to just XML. While the preferred file structure for REST is JSON, any machine-readable for may be used. There are six guiding principles for REST APIs.

The first is that client and server relationships should be loosely couple. This loose coupling allows for scalability and portability.

The second is that each request should be atomic. This ensures that the REST API remains stateless. With each request being atomic, they may be bundled and performed asynchronously. REST suggests that when doing this, a response with 202 and the URI to the status endpoint be returned to the client. This improves latency and allows the client to monitor the progress of the operations.

The third principle of REST is that responses from the API should be cacheable. Because REST utilizes HTTP, response data from HTTP GET method can be marked either cacheable or non-cacheable. In contrast, although SOAP uses HTTP as the transport protocol HTTP POST method is utilized for getting information from the web service. While it is possible to cache this data, it is much more complicated using the Response Caching Optimization Module.

The fourth principle is that the REST API should have a uniform interface. Any resource should have a single logical URI and include methods of manipulation. The resource should include links (HATEOS) to relative URIs wherever applicable.

The fifth principle is that it should be a layered/hierarchical system. Each layer can only see what is in the current layer.

The sixth principle is that the API should allow extension of functionality. This principle is arguably optional, but when it is present it allows your API to return scripts to the client.

Overall, REST is the more preferred for new implementations of web services while SOAP is more legacy. However, it is argued that while REST is generally simpler to develop, SOAP 1.2 offers a more guaranteed level of reliability and security than REST naturally does.