Universal compatibility refers to the ability of all major programming languages to read and write API messages without the need for any client SDK's. This is accomplished by 1) selecting a text format that has built-in support and 2) by not treating messages as serialized objects. Instead, API messages are flexible collections of message elements (formatted text) that are not strongly typed.

The universal compatibility of the web service API's is intended to reduce both software development costs and delivery times by eliminating the need to design, build, maintain and support client SDK's for each client programming language supported by the system.

Requirements

1. Web service API messages should be designed and built to be universally compatible for all types of applications and programming languages: depends on message-based APIs

What: *One set of web service APIs can be used by all different types of applications, written in any major programming language, without the need for client SDK's (only good documentation and good examples are needed). Avoiding the creation of a distributed-object system, and instead treating the API messages only as structured text (not as serialized objects) is the key to this level of compatibility.*

Why: *Eliminating client SDK's significantly reduces the amount of code to be written, tested and maintained. Also, this creates a clear demark between the web service API's and the client applications, simplifying maintenance, deployments, support and also reducing support costs for the API's.*

Testing: *Various tools can be used to call the API methods using different programming languages on different platforms.*

2. Web service API messages should be able to transport any type of data without the need for escaping or encoding special characters: depends on message-based APIs.

What: *All types of data, documents, files (binary), etc. should be able to be transported between the client applications and the web service methods without the need for special escaping or encoding of the data.*

Why: *If the APIs are to be used for most or all business functionality, they must have the ability to work with all types of data without any problems. Specifically, special characters in the data content must not break the structure of the API messages themselves.*

Testing: *The API Tester test harness and Postman can be used to run tests containing many different types of data as both inputs and return values to verify that they do not compromise the API message structure.*