**Web Service Miscellany**

Except in test mode, the client of either a SOAP-based or REST-style service is rarely a web browser but, rather, usually an application without a graphical user interface. The client may be written in any language with the appropriate support libraries. Indeed, a major appeal of web services is language transparency: the service interoperability – that is, the ability of web services and their consumers to interact seamlessly despite differences in programming languages, support libraries, operating systems, and hardware platforms. To underscore this appeal, my examples use a mix of languages besides Java, among them C#, JavaScript, and Perl. My sample clients in java consume services written in languages other than Java; indeed, sometimes in languages unknown.

There is no magic in language transparency, of course. If a web service written in Java can have a Python or a Ruby consumer, there must be an intermediary layer that handles the differences in data types between the service and the client languages. **XML** technologies, which support structured document interchange and processing, act as one such intermediary level. Another intermediary level is JSON (JavaScript Object Notation). XML and JSON are both data-interchange formats, but JSON clearly had the upper hand with data receivers written in JavaScript because a JSON document is the next representation of a native JavaScript object. Web service clients are increasingly JavaScript programs embedded in HTML documents and executing in a browser; such clients process JSON with less fuss than they do XML. Even among non-javaScript clients, JSON has gained in popularity; for one thing, JSON is more readable than XML because JSON has relatively less markup.