Resumo SOAP – REST

Ganhando destaque no mercado do início da década de 2000, o protocolo SOAP teve grande importância em 2003, quando passou a ser uma recomendação da W3C para desenvolvimento de serviços web, sendo o padrão mais implementado na época e deixando um legado de sistemas e integrações que perdura até hoje.

O REST, por sua vez, foi desenvolvido juntamente com o protocolo HTTP 1.1 e, ao contrário do SOAP, que tem como objetivo estabelecer um protocolo para comunicação de objetos e serviços, propôs algumas ideias de como utilizar corretamente os verbos HTTP (GET, POST, PUT, HEAD, OPTIONS e DELETE) para criar serviços que poderiam ser acessados por qualquer tipo de sistema.

Ao introduzir e adotar essas padronizações (W3C), muitos sistemas começaram a implementar esse modelo na criação de seus web services, fazendo com que o protocolo SOAP se tornasse uma das tecnologias essenciais no desenvolvimento de web services em sistemas corporativos.

No entanto, o protocolo SOAP trazia também diversas desvantagens. A primeira delas (e talvez uma das mais importantes) é o fato de que ao transportar todas as informações dentro de um envelope SOAP em XML, o conteúdo dos dados enviados de um sistema para outro se torna muitas vezes maior que o necessário, elevando tanto o consumo da banda de rede como o tempo de processamento dos dados.

Em segundo lugar, a utilização de web services SOAP não faz o uso correto dos verbos HTTP. Isso se deve ao fato de que todas as requisições SOAP são feitas através do POST de um XML, que contém o envelope SOAP.

Atualmente, a arquitetura WS-\* é composta por mais de 20 especificações. Sendo as especificações base deste conjunto a SOAP, WSDL e UDDI. Além dos citados, existem também os padrões WS-Notification, WS-Addressing, WS-Transfer, WS-Policy, WSSecurity, WS-Trust, WS-ReliableMessaging, WS-Transaction, WS-AtomicTransaction, WS-I Basic Profile entre outros. Uma das principais reclamações em relação a esse formato diz respeito a esse grande número de especificações que o torna complexo e burocrático, difícil de ser dominado por uma só pessoa.

A modelagem por trás de um serviço REST parte do princípio de seguir as boas práticas da criação de serviços HTTP e utilizar esses padrões para desenvolver web services simples e performáticos.

Um dos pontos cruciais nesse modelo arquitetural é o uso correto dos métodos disponibilizados pelo HTTP. Ao contrário do SOAP, que só utiliza o método POST para transmitir os dados, uma arquitetura REST prevê que, dentro de um cenário ideal, o método HTTP a ser utilizado seja diretamente relacionado à funcionalidade do serviço a ser consumido.