|  |  |  |  |
| --- | --- | --- | --- |
|  | SOAP teiginiai | REST teiginiai | Šaltinis (URL) |
| 1 | **Simple Object Access Protocol (*Protocol*)**  A protocol usually describes the exact messages the two(or more) peers have to exchange. Specifying also the choreography how these messages are exchanged and what they mean. | **Representational State Transfer (*Architectural style protocol)***  An architectural style does not describe messages at all, but specifies requirements (architectural constraints) that the messages, choreography or parts of the system have to fulfill. | <https://www.guru99.com/comparison-between-web-services.html>  <https://stackoverflow.com/questions/39071315/rest-is-an-architectural-style-and-is-not-a-protocol> |
| 2 | SOAP cannot make use of REST since SOAP is a protocol and REST is an architectural pattern. | REST can make use of SOAP as the underlying protocol for web services, because in the end it is just an architectural pattern. | <https://www.guru99.com/comparison-between-web-services.html> |
| 3 | SOAP can only use XML format. All data is passed through XML format. | REST permits different data format such as Plain text, HTML, XML, JSON, etc. But the most preferred format for transferring data is JSON. | <https://www.guru99.com/comparison-between-web-services.html>  <https://www.upwork.com/hiring/development/soap-vs-rest-comparing-two-apis/> |
| 4 | SOAP security is well standardized through WS-SECURITY (Web Service-Security), which is very much feature-rich and easy to implement into application code.  Also supports SSL (Secure Sockets Layer) | REST is based on HTTP – which itself is a very unsecure protocol. It supports basic authentication and communication encryption through TLS. Any further security should be additionally implemented at the server.  Also supports SSL (Secure Sockets Layer) | <https://restfulapi.net/soap-vs-rest-apis/> |
| 5 | ADVANTAGES: High security, standardized, extensibility.  DISADVANTAGES: Poorer performance, more complexity, less flexibility. | ADVANTAGES: Scalability, better performance, browser-friendliness, flexibility.  DISADVANTAGES: Less security, not suitable for distributed environments. | <https://raygun.com/blog/soap-vs-rest-vs-json/> |
| 6 |  |  |  |