SOA (Service Oriented Architecture)

* A service is a **black box** component of software
  + You care only about interfacing and how to use it not the implementation
  + You get the endpoints and you can use it
* Designing your software ecosystem and applications to work off of services
* Makes your software ecosystem easy to expand and to use
* Web Services are services available over the internet
* Services are designed to be used by machines

RESTful Web Services

* Services that adhere to the REST constraints
* REST uses HTTP(s) and only HTPP(s)

SOAP (Simple Object Access Protocol)

* It’s a type of service
* Based on XML (NOT JSON or Plain text)
  + SOAP is a bit old. Before JSON was the de-facto format for sending information over the web
  + XML was chosen because every programming language was familiar with XML and knew how to parse XML
* SOAP is not bound to HTTP. SOAP services can be created to use any protocol
* SOAP is a contract-based web service. **The WSDL is the contract**
  + WSDL 1.0 (Web Service definition Language)
  + WSDL 2.0 (Web Service description Language)
* Every SOAP service has a WSDL attached to it
  + This WSDL will tell you everything you could possibly want to know about how to use that SOAP service
  + SOAP services technically do not require documentation because the WSDL will tell you everything about that SOAP service

Tags in the WSDL

* <definitions> - The root tag of the WSDL
* <service> - Name of the soap service
  + <Port> - What endpoint the service can be used
* <binding> - lists the operations that can be performed by this SOAP service
* <operation> - A method that you can use
  + <input> - what does the method take as parameters
  + <output> - what is the return of this method
* <message> - information on an input or an output
  + <message name=”messagea”> takes in two numbers <message>
  + <message name=”messageb”> takes in a number and a String<message>
  + <input operationName = “add”>
    - Messagea
* <types>
  + Breaks down objects into primitives
  + Type:person
    - String name
    - Int age
    - String profession
* You do not send WSDLS back and forth
  + You read them once to see how to format your XML messages

|  |  |
| --- | --- |
| REST | SOAP |
| Can handle any format (usually JSON) | Exclusively XML |
| HTTP | HTTP,SMP,FTP …. |
| Endpoints designed to be intuitive to read | WSDL not super human friendly |
| Architecture design | Way to transmit information |
| No documentation not-contract based | Contract based. The wsdl serves as the ultimate authority on how to use the SOAP web service |
| Lightweight | Heavy use of XML requires more bandwidth (Takes more text to send the same information) |
|  |  |

\*It used to instead of something like FeignCLient you would feed your Java Application a WSDL which it would use to make methods to interact with the Web Service