The different between web services

1 REST operates through a solitary, consistent interface to access named resources. It’s most commonly used when you’re exposing a public API over the Internet. SOAP, on the other hand, exposes components of application logic as services rather than data. Additionally, it operates through different interfaces.

2 REST allows a greater variety of data formats, whereas SOAP only allows XM

3 REST is generally faster and uses less bandwidth

So, REST uses the http protocol fully, and it says that we already have commands in http so we will only use those, and we only need to encode the data

Then SOAP web services go and marshals the entire call.

Representational: a resource (object) can be represented in different ways (typically xml or/and json). The resource is a something on itself, but it is presented in these forms (xml, json).

State: the collection of resource (data <in the database>) manged by the service (there is a collection of resource somewhere and you put a web service in front of it to manage it),

the service itself is stateless (you call the service and you get a response, then the service forgets everything about this conversation (it might be a sort of cache somewhere otherwise no memories) means there is no session maintained between the service and the caller)

state transfer: using http commands to access and update the resource.

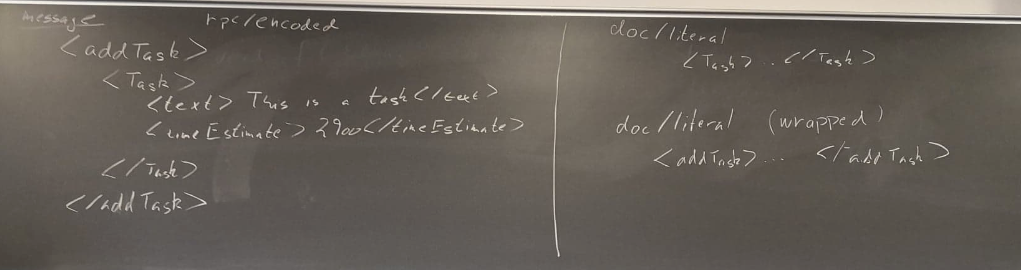
There are a lot of similarities between CRUD and HTTP commands

So, rest uses http command to perform the crud functions

Back to SOAP

So from the example we got in the class,

SOAP will do this



This is kind of a SOAP message present Add Task

This will be wrapped inside soap envelop, that says this is soap, and this is a call and now you get the message

This style called RPC/encoded, but we should add time, but Ole did not

The other style is literal. Originally in the Doc Literal style you have the idea that you do not send an encoded message…. You just send the parameter because they have a type and you just send that to web service and say here is a task deal with it. But what if there are two ways to deal with it. So when people shifted from RPC to Doc literal they say that I still need to wrap this in my task but just call it Doc literal. So, it will technically be doc literal, but it will be wrapped like RPC.

There will be differences in the WSDL, but you will essentially be ending up by sending the same xml.

Doc literal is not actually a standard.

They envelop will be slightly different, so that is one different, but the main different will be in the WSDL, in how the messages structured.

RESTful services should have following properties and features:

* Representations how to present the data
* Messages The client and service talk to each other via messages. Clients send a request to the server, and the server replies with a response
* URIs
* Uniform interface RESTful systems should have a uniform interface
* Stateless A RESTful service is stateless and does not maintain the application state for any client
* Links between resources A resource representation can contain links to other resources

Caching