Distributed System Lab3 Taeseung You

1. Cite two technologies that have avoided your need to devise and implement a protocol for communication between your client application and the server. [2 marks]

   One of the technologies is HTTP which is the protocol which can bring resources such as html files. Mostly, the browser as a client request and get responses from the server. For the data exchange of HTTP method, it's a sequence of request-response messages. Client connect to a server and server accept the connection then waits for a client's request message. Client send its request to the server and send back an HTTP response message. This method is compatatively slow with socket method however it only connect with when there's request and quit without connection so it can reduce the consumption of network width and pressure on server.

   The other technologies is that json markup language which has been used in the this part lab. Json is the open standard file format and data interchange. Json file is mostly data interchange with text file w such as xml file. Json file doesn't need conversion so that's why json file doesn't need any implementation of protocol for communication. This method is coming out because XML method has too much data because of data structure. So json method is basically based on XML method which gave XML format which give response to the request. And for json method is one of RESTful method so it controls address so this is not an actual network protocol. It sent the data using REST method without specific protocol. This method sent to URI to server through network then it returns the appropriate value from the server.

2. Explain how each of the technologies you cited fulfills the requirements of the protocol that would, otherwise, need to be implemented by you, emphasising what these requirements are. [5 marks]

   Client procedure calls the client stub in the normal way and the client stub build a message and calls the local operation system. The client's OS sends the message to the remote OS. And the remote OS gives the message to the server stub. Next, the server stub unpacks the parameters and calls the server. The server does the work and returns the result to the stub. The server stub packs the result in a message and calls its local OS. The servere's OS sends the message to the client's OS. The client's OS sends the message to the client stub. The stub unpacks the result and returns it to the client.

   For the requirement of the protocol, the protocol should go through those steps to communicate with client and server. And also few of the steps of them which are parameter marshalling and unmarshalling are the requirement of the protocol. Parameter Marshalling is that packing parameters in to a message. Marshalling is all about this transformation to neutral formats and forms, an essential part of remote procedure calls

For the HTTP side, it doesn't need any marshalling because when the library is given, we can just use request function to convert the byre data and also getting data also automatically change to client-friendly value so it doesn't need any marshalling.

Also ,in case of JSON, it doesn't need any marshalling as well when the HTTP method is given. JSON file doesn't need any unmarshalling because json file is already data contains and also as I said in first question so it doesn't need any conversion.

Basically for HTTP, the way I implement the code is that answer = self._send_request("GET",'/reservation/available') and response= requests.get(self.base_url+endpoint,headers = self._headers()). As you can see here, just using library function from requests.

And also for json file, response.json() just directly gave the value of json file.

Cited from the week 6 slides

3. Reflect on the following statement, explaining why you believe it is TRUE or why you believe it is FALSE: "Service-Oriented Architectures (SOAs) offer high flexibility and dynamicity in the construction of distributed system applications". [8 marks]

   SOA(service-oriented architecture) is a way to make software components reusable via service interfaces. SOA allows user to combine a large number of facilities from existing service to form applications

   SOA-based computing packages functionalities in to a set of interoperable services, which can be integrated in to different software systems belonging to separate business domains.

   For the SOA has Service reusability so that applications can be made from existing services. Thus, services can be reused to make many applications.

   And also easy to maintain because it has independent each other they can be updated and modified easily without affecting other services. Moreover, as I said, it is quite platform independent so making complex application by combining service from different sources and independent platform. Next, it has good availability so SOA facilities can be easily available to anyone on request.

   SOA integrates distributed and separately maintained software components. It enable by technologies and standards that facilitate component's communication and cooperation over a network. SOA is loose coupling between services. Sperate functions in to distinct services which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services communicate with each other by passing data with shared format between two or more services.

Those function thinks me SOA offer flexibility and dynamicity to construct distributed system.