

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]

REST API and HTTP.

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]

For HTTP, it defines the mechanisms for accessing and changing the data. Even though the data model of different API may be different, but getting and modifying them is the same way. For example, it provides a get method to get data from the API, post method to change the data in API and delete method to delete data in API. As a result, I do not need to write method to get and manipulate data from the server which is a requirement of client-server protocol.

For REST API, it defines a pattern for client and server communications over a network. It provides access to web service data through web URLs which are called endpoints. By using the requests method in python library, I can use the REST API to communicate with server and also passing the API key to the server to get authorisation. Moreover, it will also pass the data from the server to the client and also a HTTP response. So that I can manipulate the data passed from the server and also know the status code. If there is a miscommunication between the API and the client, I can know the problem by looking into the status code. For example, code = 200 means the requested action is successful, code = 401 means the client is unauthorised and code 500 means it has an internal server error. As a result, REST API prevents me to define a way of communication between API and client which is a requirement for the protocol and also let me know the result of the communication which is also a requirement for the protocol.

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]

I believe it is true for "Service-Oriented Architectures (SOAs) offer high flexibility and dynamicity in the construction of distributed system applications".

To begin with, it offers high flexibility in the construction of distributed system applications. As the services is completely defined by service contracts and description documents and designed as a component of SOA, we can use the same service in different distributed system applications. As a result, it is flexible for the designer to choose different service during the construction of the distribution system. For example, during the construction of a situational awareness systems, I can flexibility reuse a service to do different functionality instead of implementing a new one. Moreover, services can run on different servers within an environment, so I don't need to implement different services on different servers when developing a

distributed system application. Therefore, it is flexible for me to use the same service across different servers in constructing distributed system applications. Thirdly, each service has metadata for user to understand their functions. So, it is easier for user to understand the usage of each service and improve the flexibility by using correct service to do correct stuff. Fourthly, each service is composable. Each service can be used as building block. So, it is flexible for me to implement more complex operations by using different services as building blocks. Lastly, each service is designed to be a self-contained component in SOA. So, it has nearly no dependency on other services. This mean user can implement a service independently without the need to implement the others. As a result, user can flexibly use a service without the need to use others. All of the above characteristics of SOA give high flexibility.

Next, SOA offers high dynamicity in the construction of distributed system. Firstly, it is capable of change as service is independent from other services. So, if user want to update a service provided, they can just update this service and there is no need for updating other services that are dependent from this. Moreover, SOA service are easily available to everyone on request, so the distributed system can dynamically change from different condition. For example, it can update data from the number of users logged in. Thirdly, it is capable of action as it can handle more complex functionality. It is because as service is composable, more complex function can be build using services as building blocks. So, the distributed system can provide more complex actions by handling requests. The above characteristics give high dynamicity in the construction of distributed system.