

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]

HTTP and JSON are the two technologies used in this exercise.

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]

The requirements of the protocol needed for this exercise, would include one which allows a client to communicate with a server in the form of sending requests, and the server to communicate to the client by returning a response to such requests that is readable to the client. The two technologies HTTP and JSON both combine together to fulfill these requirements.

HTTP is a stateless protocol that meets these needs as it has HTTP methods which allows the user to specify certain requests to the server.(An overview of HTTP - HTTP | MDN, 2022) These methods, GET/POST/DELETE, means the server knows exactly how to handle requests with the given data and is able to map it to a certain URL/endpoint. In this case it allows for my code to request to, delete slots, get available/held slots, and reserve slots. The HTTP methods also return the response from the server, which can be then used to check the status codes of the request too in JSON format, so that I am able to process the response appropriately.

This reduces the need for my own implementation as I can use HTTP's functionality to communicate with the server directly and store any responses returned. HTTP also allows authorisation so it automatically handles users who are not allowed to access/edit the data from the server, again reducing the need for me to implement my own server access/restriction validation. Instead I was able to pass the authorised API token into each request in JSON format, so the server would know each request is made by an authorised client

JSON is used to transmit data between a server and client, as it has an easy-to-parse data format requiring no additional code for parsing.(What is JSON?, 2022) As mentioned before this meant that I was able to process the response from the server for every request made by the client and able to read and handle it accordingly. The JSON data returned could also be passed to the client code for it to use the JSON data for any processing. In this case I would return JSON data for available/held slots so that the client is able to find common slots for hotel and band. This functionality that JSON provides, removed the need for me to implement my own data formatter and parser, making the application much simpler.

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]

SOAs are a composition of independent services, which work together in cohesion, which allows them to be reusable via service interfaces. They can offer great flexibility and dynamicity, although this requires composition of services which is not an easy task to achieve. (Van Steen & Tanenbaum, 2017)

The main advantage of using SOAs is the service reusability, making it highly flexible, as these smaller services can be used in many applications of the system, independent of their interactions with other services. This means that certain services can be added and removed seamlessly, without causing strain on the system, and allowing for other services to remain functioning, as they are loosely coupled, meaning they do not need to know the technical details of another service it is interacting with. This leads to increased productivity as system functionality does not need to be built from scratch, instead developers can use existing code and functionality, allowing them to be more productive and save lots of time, which will also reduce costs of developing the system. (Techspirited SOA, 2022)

This reusability also leads to great reliability, as these independent services are much easier to code, test, and debug, whereas this is not the case for one large service consisting of mass amounts of code. Therefore SOAs can be relied upon to maintain a high level of quality and performance, reducing the potential for system wide errors, instead if any errors exist, they will be contained to a specific service. This therefore also makes SOAs easy to maintain, as updates and changes can easily be incorporated without conflicting with other services. These services can then be run on multiple servers at the same time with little issue, improving scalability as well which again shows how flexible and dynamic SOAs can be.

However SOAs also bring forward many challenges in trying to achieve its benefits. The main challenge is ensuring the cohesion and harmony of these composite services. This can lead to an increase in costs of developing the system, as it would require investment into technology and staff to ensure it is completed at a high standard. Depending on the type of application it might be unaffordable and unnecessary. An example of this would be a single vendor of this application, where the costs far outweighs the need for the benefits SOAs bring, instead it would be better for more suitable applications. Trying to solve these problems of harmony between services could therefore create more potential problems, and may lead to some services working whilst others cannot, meaning the development of the application would suffer as a result. (Techspirited SOA, 2022)

As well as that, once the services are working together in harmony, an increased overhead could occur as the number of services grows. This is because any data input that is passed between these services would require validation from the service, which can lead to a delay due the numerous validation processes taking place on the same data. This can eventually lead to a reduction in performance quality, making the system behave slower. This can be an issue when trying to develop a distributed system as it will slow down the process making it less flexible.

Overall I believe that SOAs do offer high flexibility and dynamicity, as the independent services allow the application to be adaptable and easily changed. Services can be added and removed, whilst also producing high quality and highly tested code. Although the multiple services can lead to an overhead which can therefore reduce overall performance of the application, the potential scalability that comes from SOAs allows it to be dynamic and highly flexible when trying to construct a distributed system application. Therefore I believe the benefits heavily outweigh the potential problems SOAs can bring.

Bibliography:

- Oracle.com. 2022. *What is JSON?*. [online] Available at: <<https://www.oracle.com/database/what-is-json/#json-use-cases>> [Accessed 10 May 2022].
- Developer.mozilla.org. 2022. *An overview of HTTP - HTTP | MDN*. [online] Available at: <<https://developer.mozilla.org/en-US/docs/Web/HTTP/Overview>> [Accessed 10 May 2022].
- Techspirited. 2022. *Advantages and Disadvantages of Service-oriented Architecture (SOA)*. [online] Available at: <<https://techspirited.com/advantages-disadvantages-of-service-oriented-architecture-soa>> [Accessed 10 May 2022].
- Book: Distributed Systems, 3rd edition (2017), by M. van Steen and A. S. Tanenbaum. Chapter 2 (pages: 63 and 64).