Critique

1. HyperText Transport Protocol (HTTP) and JavaScript Object Notation (JSON)

2. HTTP provides a specification which allows my client applications (programs) to request resources from the web servers, in this case the band and hotel servers (https://web.cs.manchester.ac.uk/band/ and https://web.cs.manchester.ac.uk/hotel/), the specification also allows the web servers to respond to these requests. An example of where this specification is used is in the reservationapi.py. In the functions reserve_slot, release_slot, get_slots_held and get_slots_available, the 'requests.GET()', 'requests.POST()' and 'requests.DELETE()' methods are used.

   - The HTTP specification allows the use of the verbs GET, POST, DELETE when web servers and client applications are communicating via the HTTP protocol over the internet.

   - The GET verb requests a representation of the of the specified resource, it only retrieves data and has no other effect.

   - The POST verb submits data to the resource, in this case the slot that my client wants to reserve as the data is included in the body of the request. Using this can result in the creation of a new resource, the update of an existing one, or both.

   - The DELETE verb is used to delete a resource and may return a representation of the removed resource.

   The JSON technology allows me to represent the responses gotten from the HTTP requests. JSON is used for the transporting and storing of data because the format is lightweight, and it is humanly readable. The JSON data is represented in name/value pairs which are separated by commas. A get request in this client for all available slots for example will return a list of "id" and slot_id pairs in curly braces to the client.

3. Service-Oriented architectures encapsulate services into independent units. Each service can be used by other services, but the services are discrete units of functionality which can be accessed remotely and updated independently. I believe the statement "Service-Oriented Architectures (SOAs) offer high flexibility and dynamicity in the construction of distributed system applications" is true. A major benefit of using this architectural style is that the services are easier to update or edit compared to larger programs that aren't split into multiple distinct services. There is also the opportunity to reuse the services instead of having to rewrite them every time they're needed in a program, this reduces the costs of testing and development. These two benefits offer more flexibility and dynamicity as they improve the agility of the enterprise or user employing the SOA style. The exchange of messages between the services in an SOA also improves the flexibility of the

system. The service in the system that handles messages can easily be changed as well as how they react to the messages they receive.