# Distributed Systems Lab 3

Horia Radu
k55592hr

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]

Two technologies are:

- HTTP
- JSON

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 HTTP technology is a very useful request-response protocol that implements client-server communication. The GET, POST and DELETE request methods to the server are a good way of making the connection from the client to the server, which otherwise, we would need to implement it ourselves.

Furthermore, the HTTP gives us the possibility of making the RPC (Remote procedure call) required to interact with the server.

The JSON technology is used for transmitting data in web applications, client to server and vice-versa. This is a very easy, readable and straightforward technology that makes it easy to transfer and translate data, for use on the client and server side. JSON also uses a minimum amount of data when transferring, thus saving time and computing power. If we wouldn't have used this technology, we would have needed to implement a way of reading, storing and transmitting data from one side to another.

One additional advantage of using JSON technology is that it is easy to use it in programming our client and server in python, as the JSON is a list of dictionaries.

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 the statement is TRUE.

Firstly, distributed system applications that encapsulate services into independent units, for which a service is a discrete unit of functionality that can be accessed remotely, via a network, and updated independently, are said to have a Service-Oriented Architecture.

SOAs, which are a composition of multiple different services, can be considered flexible based on some of their principles: autonomy, location transparency. Those principles state that the services are: autonomous, only aware of their existence and not knowing of other services and they can be called whenever and used wherever, no matter where they are present, without the need of any explicit requirements. Thus, they can respond fast and easily to changes, because they are loosely-coupled.

Basically, each service is independent, and can be accessed remotely via a network, which makes it easy to dynamically scale services independent of others.

SOAs dynamicity also results from other principles: composability, reusability and encapsulation. Services can be used in the composition of other services, they can be reused as many times as necessary and encapsulated, which means that services which were not initially planned under SOA, may get encapsulated or become a part of SOA. Also, SOA can handle data integration, while still using multiple different services and being accessed by multiple clients. All of those combined make the SOAs possible to be used in varying web applications.

But SOAs services can also be dynamically invoked and discovered, in any way a client might want.

Overall, the way the services of SOAs require to operate, work, are called and are independent between each other, makes the SOAs offer both flexibility and dynamicity.