SEMINAR 3

- What is the paper about?

In this article we can found the description of the 4 most popular APIs, describing how they work and when is the best choice to use each of them.

- What is a server/client stub, in the context of RPC?

As was mentioned in the article, a server/client stub is a piece of code that converts parameters in the context of RPC. It is all about serialization and deserialization of those parameters.

- What does it mean to be integrated with WS-security protocols? Exemplify some of these protocols and what they protect against.

In order to understand what means to be integrated with WS-security, it is important to understand the essentials. This operation took place in the context of SOAP. Soap is an XML-formatted, highly standardized web communication protocol. The WS-security integration represents one of the most important advantages of SOAP. Integrated with the WS-Security protocols, SOAP meets an enterprise-grade transaction quality. It provides privacy and integrity inside the transactions while allowing for encryption on the message level.

- How do you understand HATEOAS?

A REST API is an application programming interface that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. HATEOAS is the most mature version of REST. It serves as a vision for the long-term development of a RESTful API design. Also, loose coupling represents an important feature of HATEOAS.

- "GraphQL has *subscriptions*" - What are subscriptions? Why would we need them?

Developers use GraphQl in order to make a precise data request. Subscriptions in the context of GraphQL allow for real-time notifications from the server. Subscriptions are useful for notifying the client in real-time about changes to back-end data, such as the creation of a new object or updates to an important field. It is an important feature and we need those subscriptions when: make Small, incremental changes to large objects and when we want to have Low-latency, and real-time updates

- Order the API patterns by message size.

I would go with the following ordering by message size: RPC, GraphQl, REST, SOAP

- Which API pattern would best fit your laboratory work? Why?

If we make an abstraction of what we did in our first lab, I would prefer to go with the REST API, and here are the arguments that prove this statement: strong discoverability, good documentation, and it fits this object model well.