- What is the paper about?

The paper is about 4 API architectural styles - RPC, SOAP, REST, GRAPHQL. Describing each in detail, their advantages, disadvantages, and examples of use.

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

In the context of RPC, a stub takes care of the serialization and deserialization of the parameters.

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

WS-Security protocols provide privacy and integrity inside the transactions while allowing for encryption on the message level. Applying the mechanisms for securing web services at the message level: authentication, integrity, and confidentiality. Authentication validates the user and determines whether a client is valid in a particular context. Integrity ensures that information is not changed, altered, or lost accidentally. Confidentiality uses message encryption to ensure that no party or process can access or disclose the information in the message.

- How do you understand HATEOAS?

With HATEOAS, responses to REST requests return not only data, but also actions that can be performed on the resource. This helps make applications loosely coupled.

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

Subscriptions are a GraphQL feature that allows a server to send data to its clients when a specific event happens. Subscriptions are useful for notifying your client in real-time about changes to back-end data, such as the creation of a new object or updates to an important field.

- Order the API patterns by message size.

1.RPC, 2. GRAPHQL, 3.REST, 4. SOAP

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

Any can be applied, but the “suitable” will be the use of REST, because it has the highest level of abstraction and best modeling of the API. It permits to have decoupled client and server, achieving flexibility and remaining stable which is great for distributed systems. And the microservices in my laboratory work are based on CRUD operations supported by REST API.

Link to the paper:
https://levelup.gitconnected.com/comparing-api-architectural-styles-soap-vs-rest-vs-graphql-vs-rpc-84a3720adefa