Distributed Systems (H0N08A) 2023-2024

1 | Remote communication
Report

For the two remote communication lab sessions, we require you to submit a report where you reflect on the studied technologies and how they compare. You should write this report in groups of *two* people. Provide your answers in English, and keep them short, concise and to the point: a few lines is often enough. You are encouraged to use the LaTeX template on Toledo. Don't forget to put your names in the report.

Submission. Before Friday **20 October** at 19:00, **each** student must submit this report to Toledo. Submit these to Toledo under `Assignments > Submission 1: Remote communication`.

Questions. Compile a report that answers the following questions:

1. What is the role of stub and skeleton in Java RMI? Does REST have similar concepts?
2. What is the difference between serializable and remote? In your Java RMI assignment, which classes were made serializable and which classes were made remote? Motivate your choice.
3. What role does the RMI registry play? Why is there no registry in REST?
4. How do you make sure that your Java RMI implementation is thread-safe? Moreover, which methods were required and selected to be thread-safe? Can you motivate your decisions?
5. Level 3 RESTful APIs are hypermedia-driven. How does this affect the evolution of your software, and more specifically your APIs in terms of coupling and future upgrades?
6. During the REST session, you used a code generator to generate the server side of the application. What other components could you generate from the OpenAPI specification?
7. Your OpenAPI specification indicates that "Name" is a mandatory field for a meal. Your client received the specification, can you be sure "Name" will always be present when you receive a request from that client? When you generate server code from this specification, will your application check if "Name" is always present?
8. What is the advantage of using code generation, e.g. using OpenAPI, over a language-integrated solution such as Java RMI? What are the downsides of using code generation from an implementation perspective?
9. How do Java RMI and a RESTful service compare in terms of flexibility (e.g., at platform level), extensibility (in particular when working with third parties), and susceptibility to protocol errors?
10. Suppose that you are tasked with developing the following applications, which of the two remote communication technologies (Java RMI, REST) would you use to realize them? What is your motivation for choosing a specific technology? If you choose REST, is it beneficial to also use OpenAPI?
    (a) A public web API that can be used by applications all over the world.
    (b) The internal communication of a high-performance SaaS application that consists of multiple distributed components, written in various programming languages.
    (c) An internal distributed application of a large company, written in Java only.