Distributed Systems (H0N08A)

# *Report:* JAVA RMI

Dries Janse *(r0627054)*
Steven Ghekiere *(r0626062)*

October 29, 2019

**1. How would a client complete one full cycle of the booking process, for both a successful and failed case? Base yourself on the example scenarios in Figure 1. Create sequence drawings to illustrate this.**
Hier gaan we een sequence d

**2. When do classes need to be serializable? You may illustrate this with an example class.**
When a Java class implements the Serializable interface, an instance of this class can be passed as a result or an argument in Java RMI. These instances, just as all primitive types, are copied and passed by value. This means that the receiver creates a copy of the object. On this new copy object, methods can be invoked but this will only change the local object. The state of the local object can be different from the state of the original object of the sender.
Classes need to be serializable when the value of the object of such a class is needed. The users (clients) of these classes are not allowed to modify the original objects. In our project, we made the following classes explicitly serializable:

CarType

- ReservationConstraintes
- Quote
- Reservation

The following classes, used in our project, are already serializable: String, Date, HashSet, ArrayList and HashMap.

**3. When do classes need to be remotely accessible (Remote)? You may illustrate this with an example class.**
Answer 2

**4. What data has to be transmitted between client and server and back when requesting the number of reservations of a specific renter?**
Answer 2

**5. What is the reasoning behind your distribution of remote objects over hosts? Show which hosts execute which classes, if run in a real distributed deployment (not a lab deployment where everything runs on the same machine). Create a component/deployment diagram to illustrate this: highlight where the client and server are.**

Answer 2

**6. How have you implemented the naming service, and what role does the built-in RMI registry play? Why did you take this approach?**
Answer 2

**7. Which approach did you take to achieve life cycle management of sessions? Indicate why you picked this approach, in particular where you store the sessions.**
Answer 2

**8. Why is a Java RMI application not thread-safe by default? How does your application of synchronization achieve thread-safety?**
Answer 2

**9. How does your solution to concurrency control aspect the scalability of your design? Could synchronization become a bottleneck?**
Answer 2