

Report: Java RMI

Thomas Bamelis (*r0640219*)

Michiel Jonckheere (*r0665594*)

October 29, 2019

Question 1 TODO Michiel, diagram

The client looks up information he needs for making a quote. When a client make a quote with certain constraints, it asks for a quote to the car rental agency who returns the first available quote at one of the registered car companies.

The client can do this for a number of quotes, which are all saved to his session. When the client wants to confirm his quotes, he notifies the car rental agency. The car rental agency tries to confirm all quotes at the relevant car rental companies. If one fails, the already confirmed reservations are cancelled and the client is notified of the failure.

Question 2 TODO wa mooi typen Michiel Alst over de draad moet basicly, zoals een CarType bijvoorbeeld When it is either a parameter or a return value of a remote method (so also when it is contained in an object which has to be serializable).

Question 3 When a remote process wants to call a method on a local object. als ze methods bevatten die remotely callable zijn. de interface extends van Remote, de klasse implementeerd die, zoals ICarRentalAgency en CarRentalAgency.

Question 4 TODO Michiel proper typen parameters en return values: clientName en the return integer gewoon kijken wat er naar server meegegeven wordt en wat er terug gegeven wordt?

Question 5 First the client CRA communication:

The client and CRA are on a different server. The client uses a CRA-stub which it looks up on a registry on the CRA-server to receive manager and reservation session-stubs. Interacting with the CRA is then done through this sessions. Every session on the client is a remote interface for a session object on the CRA-server. The session object then interacts locally with the CRA-object on the CRA-server.

All interaction run through the CRA-object and the client never directly talks to a CRC.

The CRA interacts with different CRC's which are all on their own remote servers. The CRA uses an interface to a remote CRC-object when interaction is necessary. The CRA keeps a list of interfaces of the registered CRC's.

The only registry is located on the CRA and everybody interacts with it.

When a CRC wants to register itself it pushes itself to this remote registry and notifies the CRA through a managersession.

Question 6 The CRA uses a hashmap with as key the name of the company and as value an interface stub to the remote object. A CRC pushes itself on to the registry before notifying the CRA with the name by which it can lookup the interface.

Question 7 When a session is created it is created on the CRA-server and it pushes itself to registry. The CRA-server then sends the lookup name to the client for it to be used. When the client is done with the session, he calls close session on the remote interface to the remote object. The remote object then removes itself from the registry, thus removing the last local reference to it causing the garbage collector to collect the session.
Sessions are stored on a CRA-server ..TODO

Question 8 opzoeken eerste deelvraag

When a write is done, which is a change to a CRC-object, the CRA blocks any other call. So it's impossible for a bad read to happen this way. Blocking is only necessary when quotes are being confirmed.

Question 9 block van synchronized, alles van de klasse CarRentalAgency moet wachten als createQuote en confirmQuote opgeroepen worden. Anders kan er verkeerde info doorgegeven worden met de andere methods, bijvoorbeeld checkAvailableCarTypes. Bij het doorgeven van verkeerde info kan dan een foute quote gecreeerd worden. deze zal resulteren in een error, en zal de checkAvailableCarTypes terug opgeroepen moeten worden

The scalability is "impacted" in that only one client can confirm at a time and while confirming no other clients can do something. When there would be a lot of users, this could be a potential bottleneck.