RMI

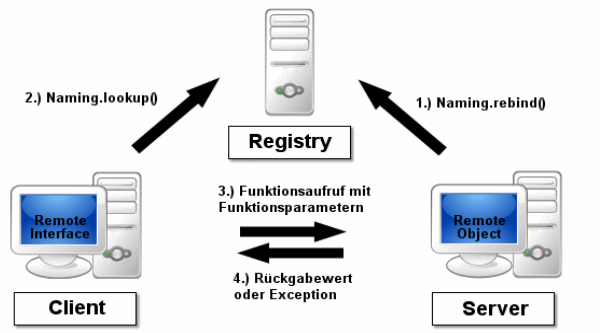
- RMI – Remote Methode Invocation
- Client / Server architecture
- Kommunikation via Objektreferenzen oder mit Hilfe von Serialisierung

Aufbau

- Server: skeleton(Remote object) implements *Remote* – Interface
- Client: stub (Stellvertreter des Remote Objects)

Funktion

- Server meldet Skeleton mit eindeutigem Namen bei der Remote Registry an. *Naming.bind/rebind()*
- Client fragt Namen ab, bekommt Objektreferenz (oder Objektkopie)und erzeugt dadurch den stub. *Naming.lookup()*
    - Kopie, wenn *serializable* sonst Referenz
- Client ruft an Stub Methode auf. Server führt Methode aus und schickt nur Ergebnis zurück.



Serialisierung

- Kapselung von Objektgeflechten (Zustand und interne Referenzen) zu einem linearen Bytestrom. Deserialisierung zur Stub-Erzeugung (falls kein Remote-Object).
- Übertragungsmechanismus von RMI (*call by value*)

UI – Threading

- EDT – Event Dispatch Thread: erledigt alle UI-Berechnungen
- Nicht nebenläufigkeitssicher → AWT.Eventqueue
- Nicht für andere Berechnungen als UI → *Swingworker* oder *Timer* als Background-Thread

Sources

- http://www.ti.uni-tuebingen.de/fileadmin/assets/csp_ws0809/aufgabe2/aufgabe2a.pdf
- http://www.javacoffeebreak.com/articles/javarmi/javarmi.html
- http://de.wikipedia.org/wiki/Remote_Method_Invocation
- http://www.oreilly.de/catalog/javarmi/chapter/
- http://patriot.net/~tvalesky/easyrmi.html
- http://www.cs.swan.ac.uk/~csneal/InternetComputing/RM2.html
- http://java.sun.com/products/jfc/tsc/articles/threads/threads1.html