

Mobile und Verteilte Datenbanken

Java RMI

Vorlesung
Wintersemester 2009/2010

Sven Groppe groppe@ifis.uni-luebeck.de



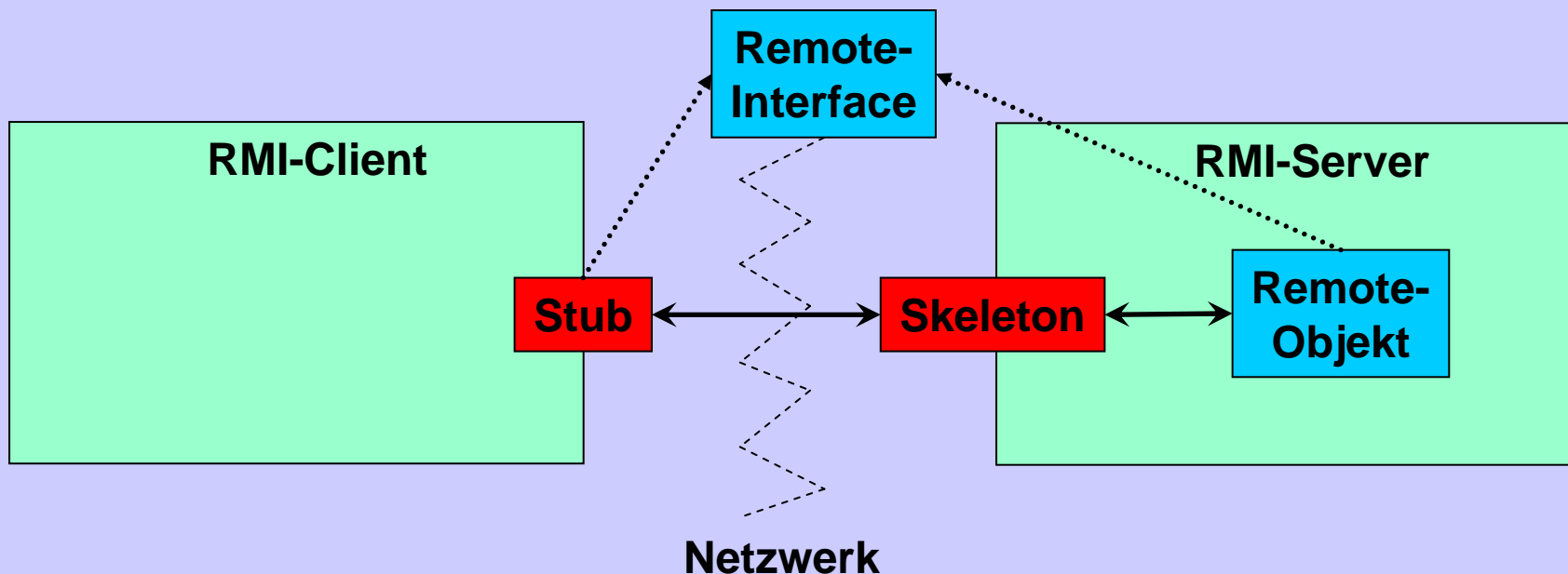
Institut für Informationssysteme



Universität zu Lübeck

Java Remote Method Invocation (RMI)

- Realisierung von
 - verteilter Kommunikation und
 - Methodenaufrufe entfernter Objekte möglich



Remote-Interface, Stub und Skeleton

- Remote-Interface vereinbart Methoden zwischen Client und Server
z.B.:

```
public interface ServerInterface extends Remote {  
    public String transmitBinding(Bindings b) throws RemoteException;  
}
```

- Remote-Objekt implementiert Remote-Interface
- Client-Stub
 - implementiert das Remote-Interface
 - erledigt Kommunikation zwischen Client-Stub und Server-Skeleton auf Seite des Clients
- Server-Skeleton
 - erledigt Kommunikation zwischen Client-Stub und Server-Skeleton auf Seite des Servers
 - ruft Methoden des Server-Objektes auf
- Client-Stub und Server-Skeleton werden „automatisch“ generiert

Server: Remote-Objekt und Java-Registry

- Bereitstellung der Server-Funktionalität
- implementiert Remote-Interface
z.B.:

```
public class ServerProc extends UnicastRemoteObject implements ServerInterface {  
  
    protected ServerProc() throws RemoteException { super(); }  
  
    public String transmitBinding(final Bindings b) throws RemoteException {  
        System.out.println("Bindings retrieved:");  
        System.out.println(b.toString());  
        return "Server retrieved the bindings " + b.toString(); }  
}
```

- muss unter einem Namen bei der Java-Registry auf dem Server gebunden werden:
z.B.:

```
try {  
    final Registry reg = LocateRegistry.createRegistry(Registry.REGISTRY_PORT);  
  
    final ServerProc obj = new ServerProc();  
    // Bind this object instance to the name "serverXSLTProc"  
    Naming.rebind("serverProc", obj);  
  
} catch (final Exception e) { System.out.println(e.getMessage()); }
```

Client

- Lookup des Server-Objektes und Erhalten des Stubs, der das Server-Interface implementiert
- Danach Methodenaufrufe auf Stub-Objekt

z.B.:

```
try {
```

```
    ServerInterface stub =
```

```
        (ServerInterface) Naming.lookup("//pc18/serverProc");
```



Hat die Form: **//{hostname|ipaddress} [:port]/objectname**
(anderes Bsp.: **//pc18.pool.ifis.uni-luebeck.de:1099/serverProc**)

```
System.out.println("Receiving text "
```

```
    + serverInterface.transmitBinding(Bindings.createNewInstance());
```

```
} catch (final Exception e) { System.out.println(e.getMessage()); }
```