

# Konzepte von Betriebssystem-Komponenten

Middleware

RMI

Mario Kiefer

21. Januar 2005

# Inhalt

- 1 Einführung
- 2 Architektur von RMI
- 3 Grundlagen
- 4 Naming Service
- 5 Beispiel

# Inhalt

- 1 Einführung
- 2 Architektur von RMI
- 3 Grundlagen
- 4 Naming Service
- 5 Beispiel

# Was ist RMI ?

- RMI = Remote Method Invokation
- Java Pendant zu RPCs
- Seit Java 1.1 im Sprachumfang enthalten
- Dient zur Erstellung von verteilten Anwendungen

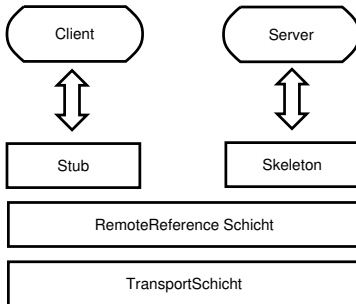
# Bestandteile von RMI

- Paket java.rmi
- RMI Compiler rmic
- RMI Naming Service rmiregistry
- RMI activation system daemon rmid

# Inhalt

- 1 Einführung
- 2 Architektur von RMI**
- 3 Grundlagen
- 4 Naming Service
- 5 Beispiel

# Architektur von RMI



# Inhalt

- 1 Einführung
- 2 Architektur von RMI
- 3 Grundlagen**
- 4 Naming Service
- 5 Beispiel

# Serialisation

- Mechanismus um Objekte in einen Bytestream umzuwandeln
- Objekte müssen Marker Interface Serialisation implementieren
- Primitive Datentypen sind bereits serialisierbar

# Parameterübergabe

- Call-by-Reference nur bei Remote-Objekten
- Call-by-Value bei Objekten und primitiven Datentypen
- Methodenaufruf bei Empfänger bezieht sich nur auf lokale Kopie

# Grundlagen

- Garbage Collection  
Referenzzähler auf Remote-Objekt
- Dynamisches Laden von Klassen  
Eigenschaft `java.rmi.server.codebase`
- RMI SecurityManager  
Erlaubt es Client und Server eine Verbindung aufzubauen  
bzw. Klassen nachzuladen

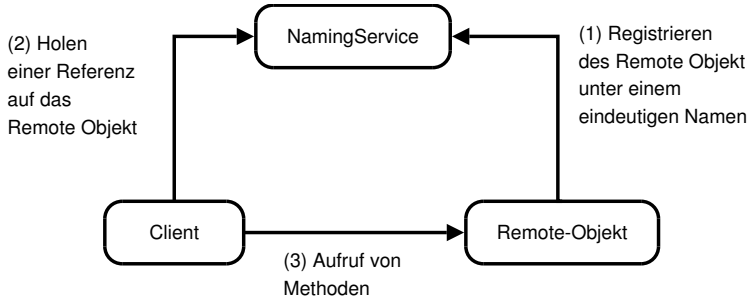
# Inhalt

- 1 Einführung
- 2 Architektur von RMI
- 3 Grundlagen
- 4 Naming Service**
- 5 Beispiel

# Naming Service

- Verknüpft Remote-Objekt mit einem Namen
- Zentrale Auskunftsstelle für Client
- Server registriert seine Remote-Objekte

# Beispiel Ablauf



# Inhalt

- 1 Einführung
- 2 Architektur von RMI
- 3 Grundlagen
- 4 Naming Service
- 5 Beispiel**

## Beispiel

# Vorführung

# Fragen

Fragen ?