

Client – Server -Anwendungen mit UML und Java

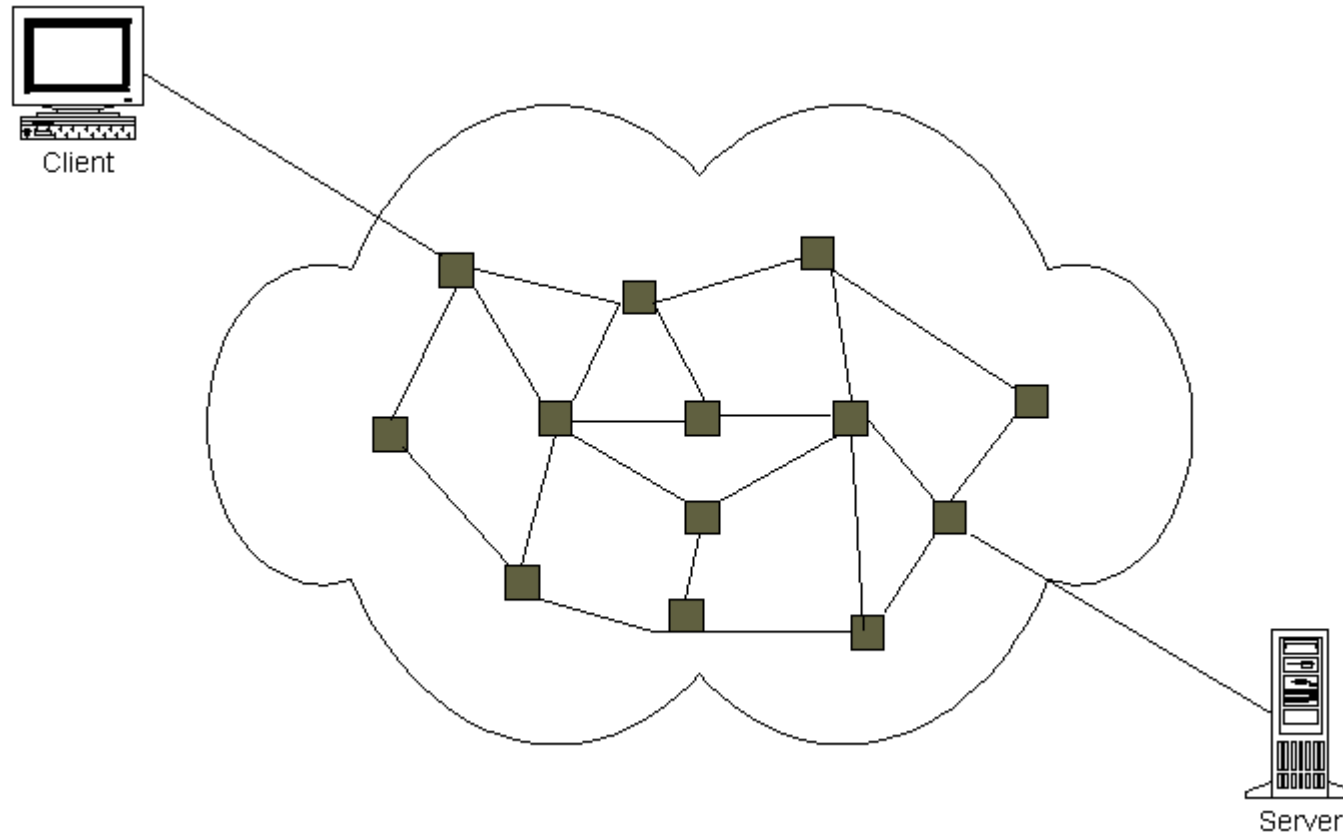
3. Informatiktag NRW – 29.3.04

Barbara Leipholz-Schumacher
Euregio-Kolleg, Würselen

- Netzwerkgrundlagen
 - Ports und Sockets
 - Protokolle
 - http
 - smtp
- Client-Server-Anwendungen mit Sockets
 - Echo-Server und -Client
 - Wahl-Server und -Client
 - Datei-Server und -Client
 - Ein Reservierungssystem
- Client-Server-Anwendungen mit RMI
 - Eine Online-Bank
 - Ein Chat

Netzwerkgrundlagen

Ermittlung der Route



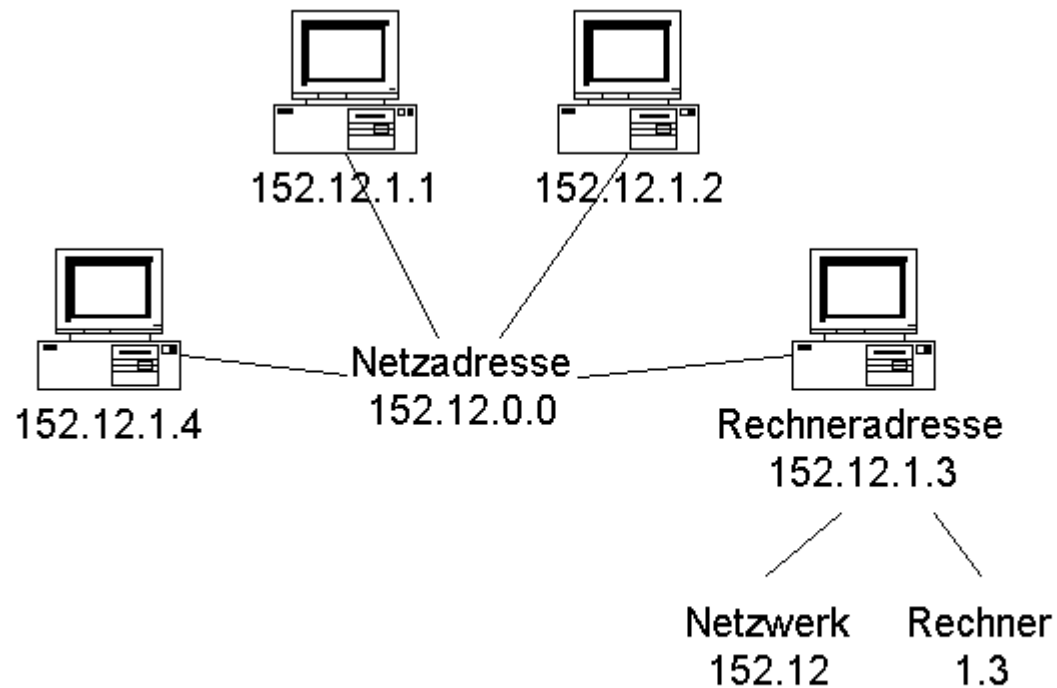
Mail an den europäischen Schulserver info@www.eun.org

- Linux: traceroute
- Windows: trace
- WWW: www.visualroute.de

traceroute to www.eun.org (130.226.1.180), 30 hops max, 40 byte packets

```
1 cisco-schule.rz.RWTH-Aachen.DE (137.226.75.129) 21.890 ms 21.712 ms 21.676 ms
2 cisco.rz.RWTH-Aachen.DE (134.130.79.254) 22.595 ms 22.495 ms 22.564 ms
3 cisco-rz.rz.RWTH-Aachen.DE (137.226.144.1) 23.003 ms 22.760 ms 24.312 ms
4 RWTH-Aachen1.WiN-IP.DFN.DE (188.1.6.29) 30.267 ms 23.622 ms 22.939 ms
5 ZR-Koeln1.WiN-IP.DFN.DE (188.1.160.17) 27.136 ms 24.897 ms 23.778 ms
6 ZR-Frankfurt1.WiN-IP.DFN.DE (188.1.144.34) 29.792 ms 28.225 ms 28.199 ms
7 IR-Frankfurt1.WiN-IP.DFN.DE (188.1.164.22) 29.900 ms 29.610 ms 31.808 ms
8 dfn.de.ten-155.net (212.1.192.133) 32.408 ms 33.633 ms 31.183 ms
9 de-se.se.ten-155.net (212.1.192.66) 71.511 ms 71.127 ms 66.176 ms
10 sw-gw.nordu.net (212.1.192.154) 72.142 ms 69.661 ms 72.685 ms
11 e-gw.nordu.net (193.10.252.209) 76.529 ms 80.833 ms 81.334 ms
12 dk-gw.nordu.net (193.10.252.94) 90.209 ms 87.870 ms 89.813 ms
13 eunwww1.uni-c.dk (130.226.1.180) 100.113 ms * 81.031 ms
```

Aufbau der IP-Adressen



Jede IP-Adresse ist 4 Byte lang und identifiziert Netzwerk und Rechner

Klasse A

26 00011010	104	0	19
8 Netzwerkbits	24 Rechnerbits		

Klasse B

128 10000000	66	12	1
16 Netzwerkbits	16 Rechnerbits		

Klasse C

194 11000010	153	13	2
24 Netzwerkbits	8 Rechnerbits		

- Beispiel:** Die IP-Adresse 134.99.128.75 klassifiziert ein Klasse B - Netz, denn die Dezimalzahl 134 besitzt die duale Darstellung 10000110, beginnt also mit 10. D.h.
- die ersten beiden Byte legen die Netzadresse fest: 134.99.00.
 - die Rechneradresse ist 128.75.

Subnetze

Durch Festlegung der Subnet-Maske kann die Netzadresse verlängert werden.

Die Subnet-Maske besteht ebenfalls aus 4 Byte, jedes Byte hat i.a.

- entweder den Wert 0 (binär 00000000)
- oder 255 (binär 11111111)

Die übereinanderstehenden Bit werden mit dem logischen UND miteinander verknüpft:

Subnet-Maske	11111111	11111111	11111111	00000000
IP-Adresse	10000110	01100011	10000000	00010011
Netz + Subnet-Adresse	10000110	01100011	10000000	00000000
Rechneradresse				00010011

- Die ersten beiden dualen Ziffern 10 klassifizieren ein Klasse B Netz.
- Die Subnet-Maske 255.255.255.0 legt fest, daß auch das 3. Byte zur Netzadresse gehört und nur das 4. Byte die Rechneradresse ist.
- Netz-Adresse ist also 134.99.128.0, die Rechner-Adresse ist 75.

Freie IP-Adressen

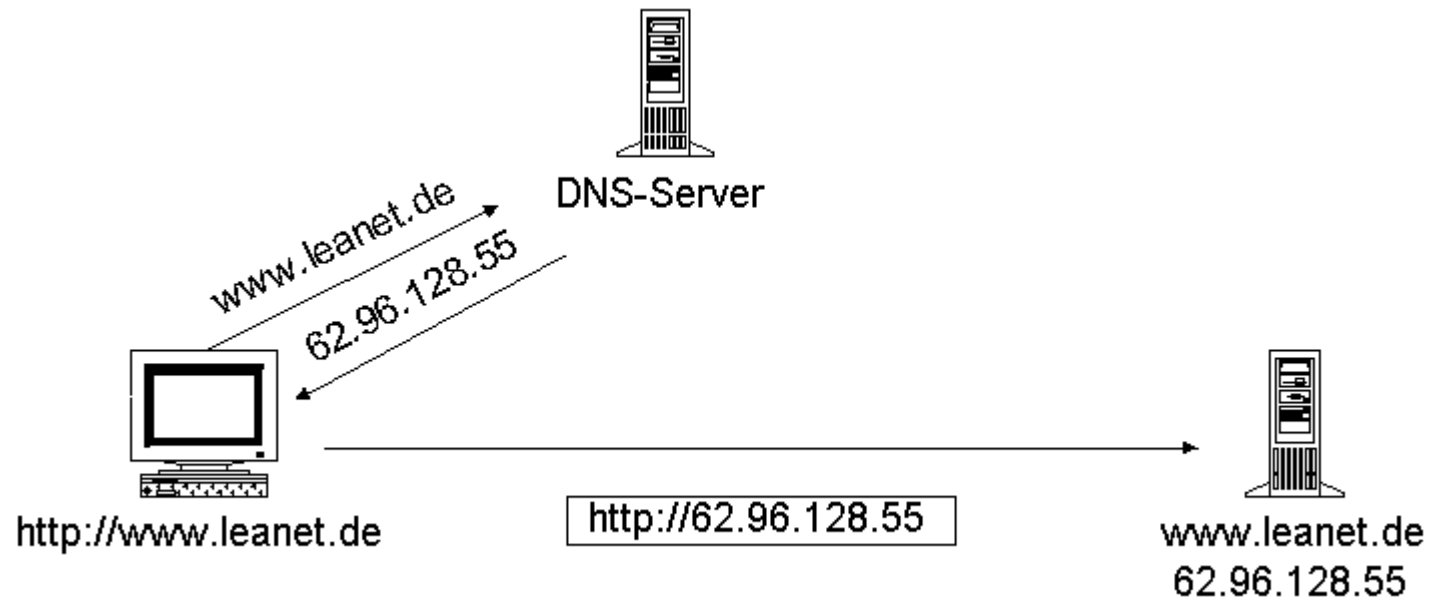
Diese Adressen sind nur im lokalen Netz bekannt.

Class A	10.0.0.0	bis	10.255.255.255
Class B	172.16.0.0	bis	172.31.255.255
Class C	192.168.0.0	bis	192.168.255.255

Sonderfall - eigener Rechner

localhost: 127.0.0.1

Zuordnung einer IP-Adresse zu einem Namen

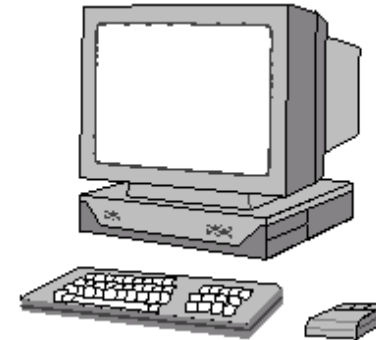
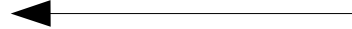


Client – Server - Modell

Server antwortet



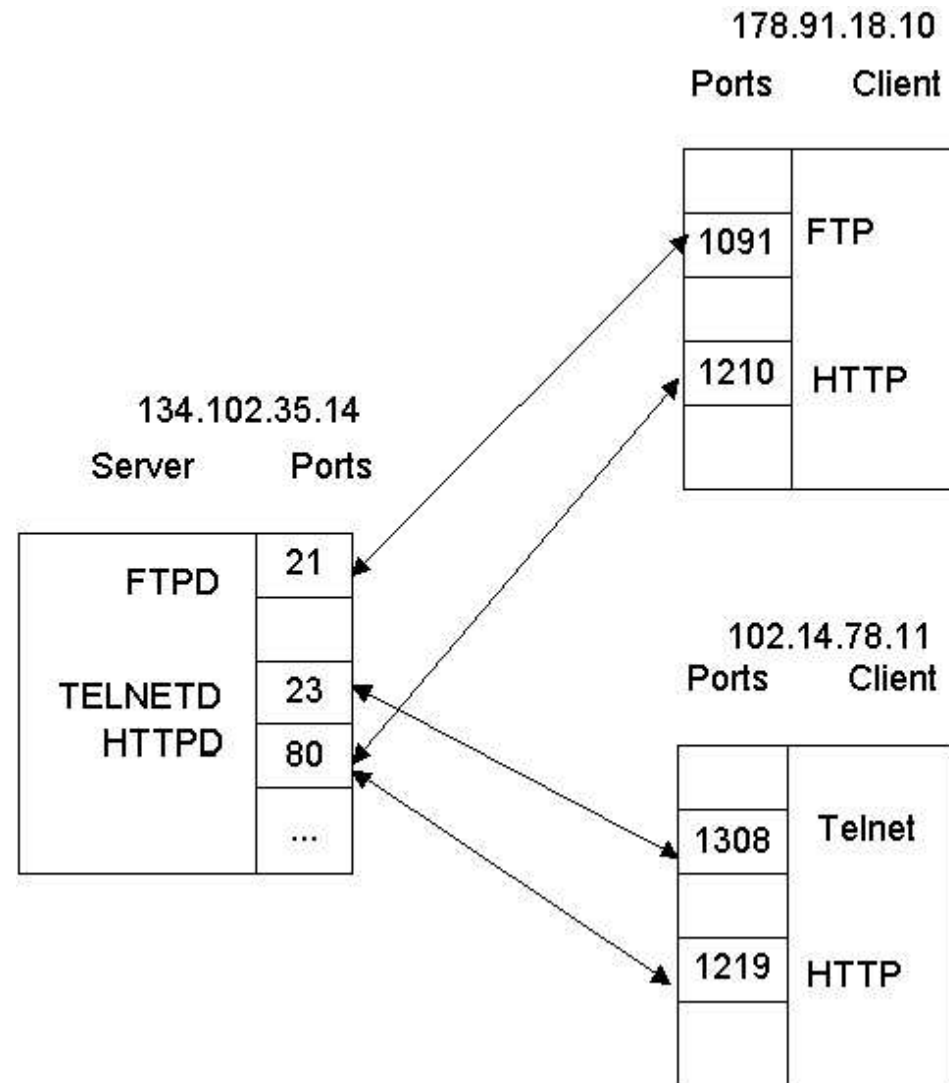
Client fragt

<http://www.google.de>Sendet das gewünschte
WWW-Dokument

Dienste:

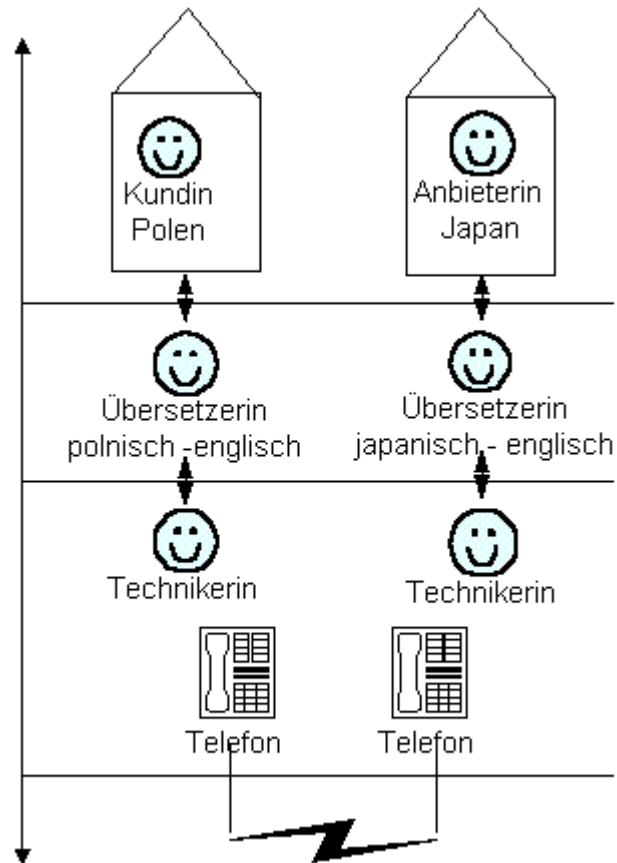
- **http** – **H**ypertext **T**ransfer **P**rotokoll
- Auf einem entfernten Rechner arbeiten
 - **telnet**
 - **ssh** – **S**ecure **S**hell, Daten werden verschlüsselt übertragen
- **ftp, sftp** – **S**ecure **F**ile **T**ransfer **P**rotokoll
- **smtp** – **S**imple **M**ail **T**ransfer **P**rotokoll
- ...

Ports und Sockets



Protokolle

Schichtenmodell



Jede TeilnehmerIn empfindet die Kommunikation als horizontal

- Die tatsächliche Kommunikation ist mit Ausnahme der Schicht 1 vertikal
- Die Schichten sind voneinander unabhängig, davon ausgenommen sind ihre Schnittstellen
- Der Inhalt der Kommunikation ist nur für die beiden Geschäftsfrauen interessant, alle anderen bleiben davon unberührt
- Die ÜbersetzerInnen sind frei in der Wahl ihrer Sprache, alle anderen bleiben davon unberührt
- Keine der unteren Ebenen interpretieren die Daten, sie formen sie lediglich um

<p style="text-align: center;">OSI</p> <p style="text-align: center;">Open Systems Interconnect Reference Model</p>	<p style="text-align: center;">TCP/IP</p>
<p>7. Anwendungsschicht besteht aus Anwendungen, mit denen man das Netz nutzen kann</p> <p>6. Darstellungsschicht standardisiert das Format der Daten auf dem Netz</p>	<p>4. Anwendungsschicht enthält Anwendungen und Prozesse, die auf das Netzwerk zugreifen</p>
<p>5. Kommunikationssteuerungsschicht verwaltet die Verbindungen zwischen den Anwendungen</p> <p>4. Transportschicht garantiert die fehlerfreie Datenübertragung durch Fehlererkennung und Korrektur</p>	<p>3. Transportschicht stellt Rechner-zu-Rechner-Datendienste zur Verfügung</p>
<p>3. Vermittlungsschicht verwaltet die Verbindungen zwischen den Rechnern im Netz für die höheren Schichten</p>	<p>2. Internetschicht definiert den Aufbau von Datagrammen und routet Daten durchs Netz</p>
<p>2. Sicherungsschicht sorgt für die zuverlässige Übertragung der Daten über die physikalischen Verbindungen</p> <p>Bitübertragungsschicht definiert die physikalischen Eigenschaften der Übertragungswege</p>	<p>1. Netzzugangsschicht enthält Routinen für den Zugriff auf physikalische Netze</p>
<p>1. Physikalisches Übertragungsmedium</p>	

Anwendungsschicht - das http-Protokoll

Anfrage des Client :

http://	www.die.et-inf.uni-siegen.de	/e-learning/iml/is/rvs/	lokalene.html
Das Protokoll	der Name des WWW-Servers im Internet	Der Pfad auf der Festplatte des Servers	der Dateiname

gesendeter Text	Erläuterung
Get /iml/is/rvs HTTP1.0	mit GET fordert der Browser die Seite an, zusätzlich gibt er an, welche HTTP-Version er versteht
Accept: text/html	spezifiziert die Datenformate, die er bevorzugt
User-Agent: Mozilla/4.77	Browser des Client und Versionsnummer
Host 135.206.14.23	IP-Adresse des Client
...	

Der Server antwortet

- mit Informationen über sich selber und das angeforderte Dokument
- dem Inhalt des Dokumentes **sofern es vorhanden ist.**

Anwendungsschicht: Das smtp-Protokoll

telnet servername 25

telnet turing.euregio-kolleg.de 25

Trying 192.168.100.7...

Connected to turing.euregio-kolleg.de.

Escape character is '^]'.

220 turing.euregio-kolleg.de ESMTP Sendmail 8.8.8/8.8.8; Thu, 27 Jan 2000 21:08:27 +0100

helo turing

250 turing.euregio-kolleg.de Hello barbara@ada.euregio-kolleg..de [192.168.100.15], pleased to meet you

mail from:tux@linux.de

250 tux@linux.de... Sender ok

rcpt to:sonja@euregio-kolleg.de

250 sonja@euregio-kolleg.de... Recipient ok

data

354 Enter mail, end with "." on a line by itself

Hallo Sonja,

...

Viele Gruesse von Tux

.

250 VAA00724 Message accepted for delivery

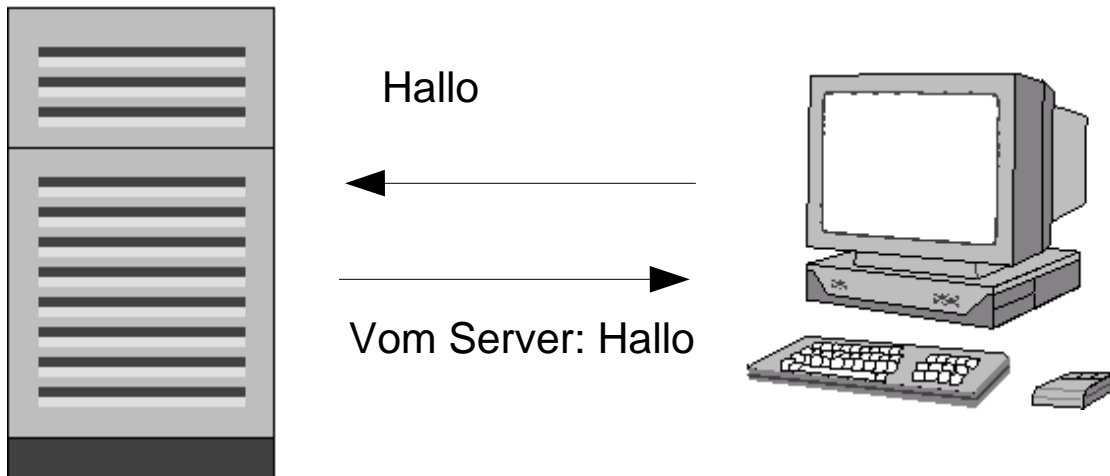
quit

Client-Server-Applikationen mit Sockets

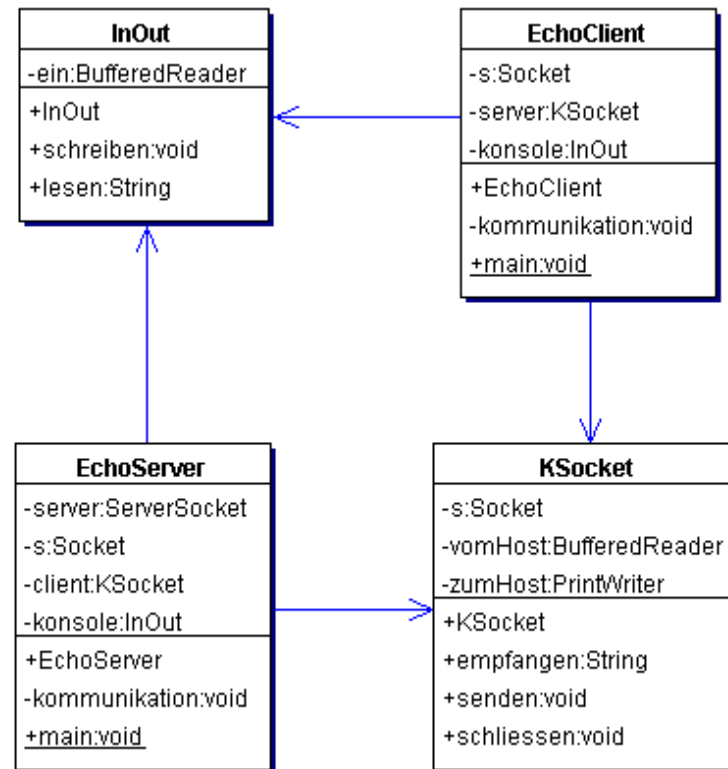
Echo-Server

Server antwortet

Client fragt



Echo-Server – UML



1. Start des Servers
2. Start des Client

Fehleranalyse mit Telnet

Im DOS-Fenster:

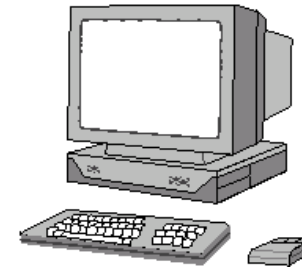
```
telnet server port
```

```
telnet 192.168.0.91 1100
```



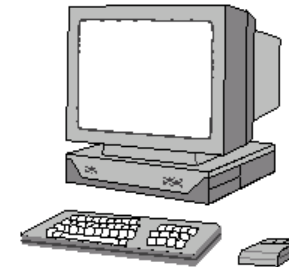
Hallo

Vom Server: Hallo



Wie gehts

Vom Server: Wie gehts

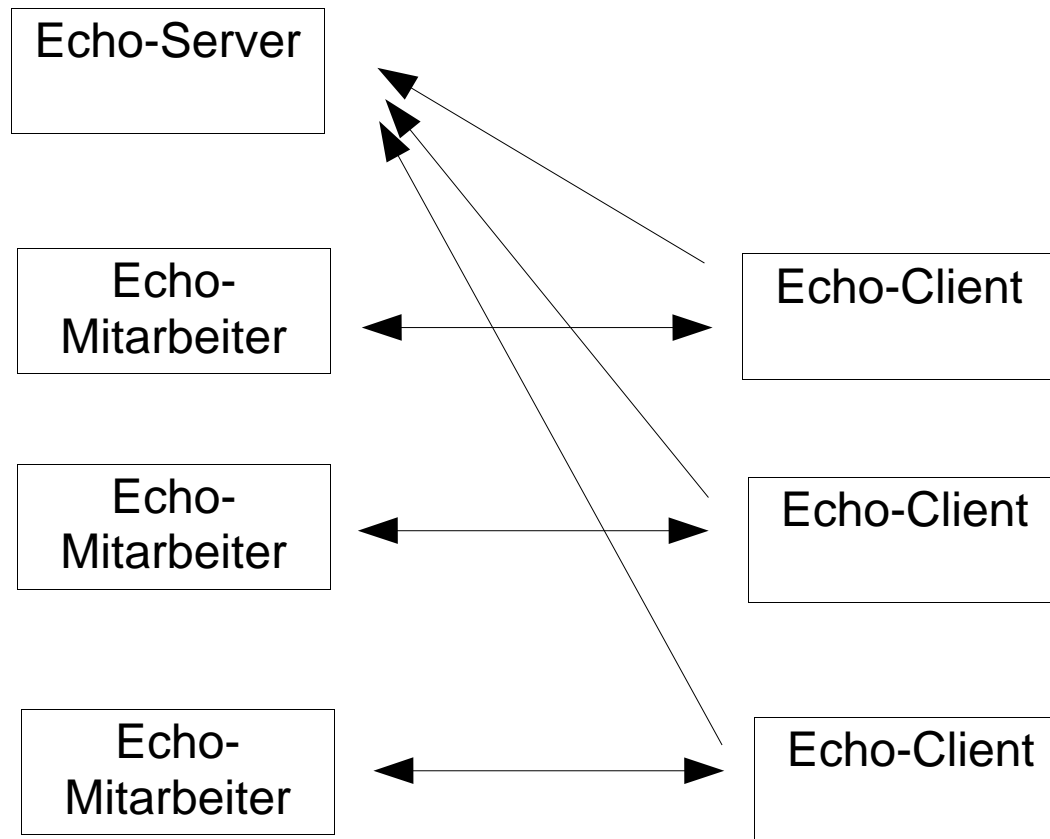


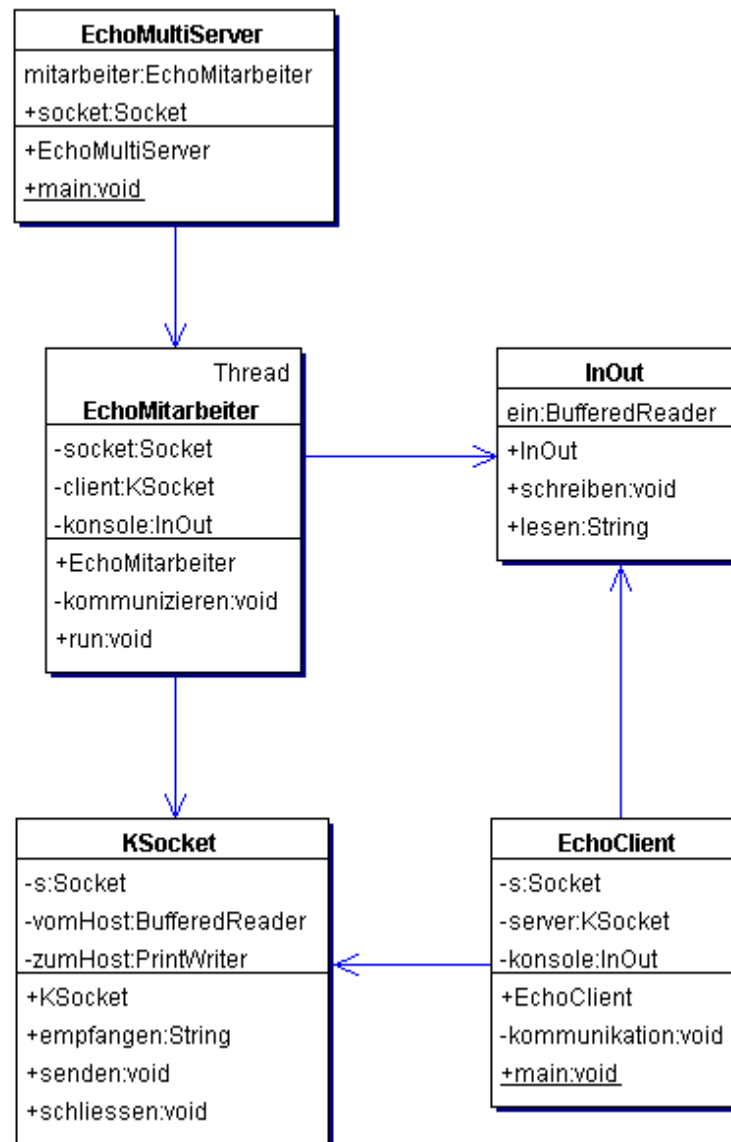
bye



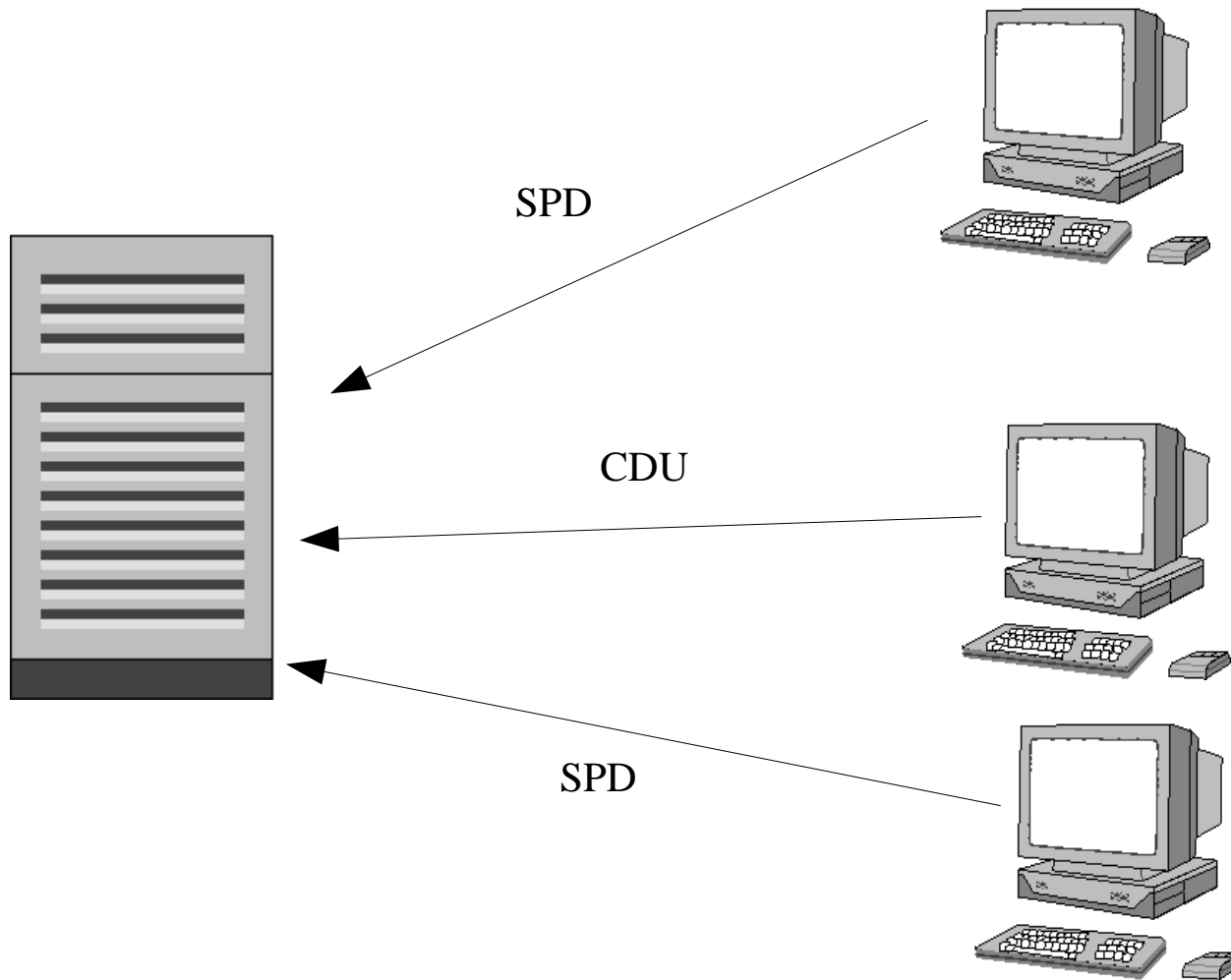
Echo-MehrfachServer

Der Echo-Server startet
für jeden Client einen
Mitarbeiter

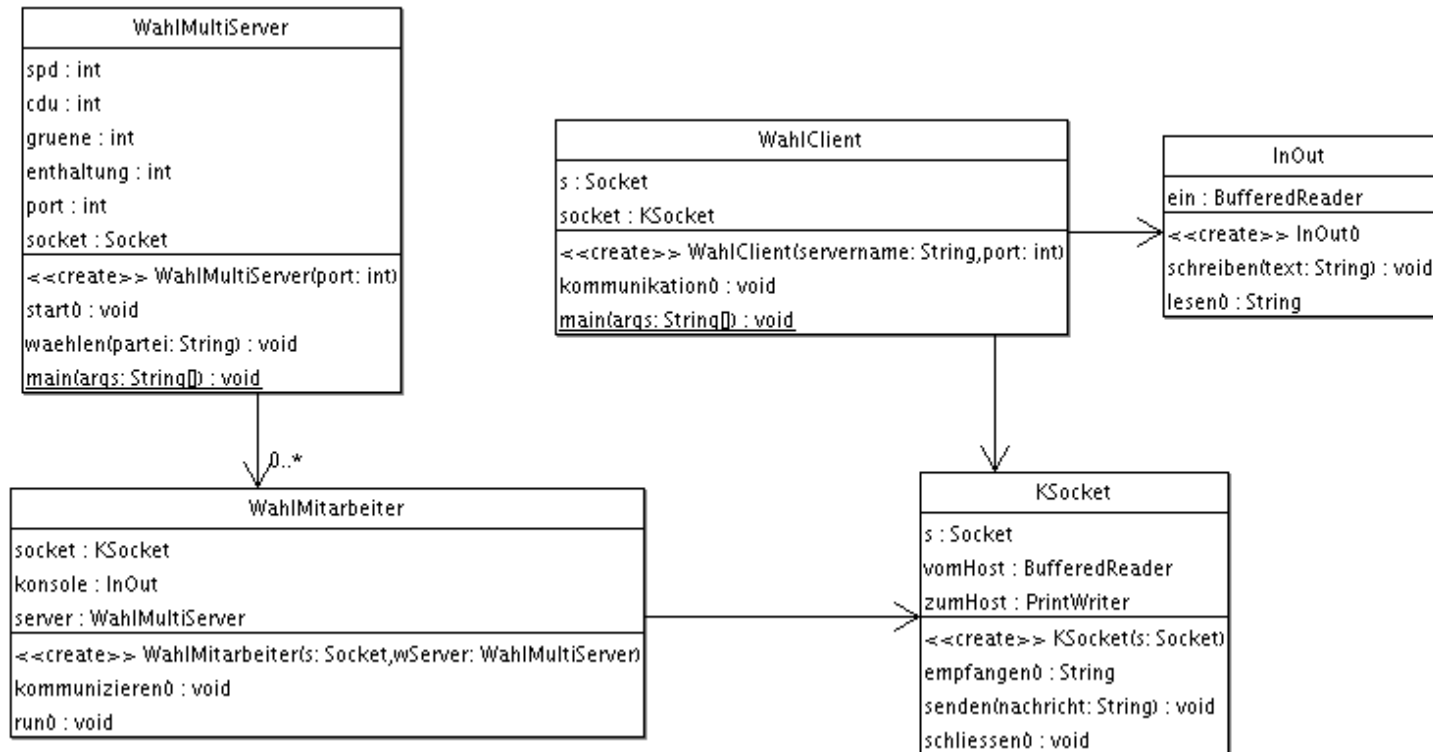




Wahl



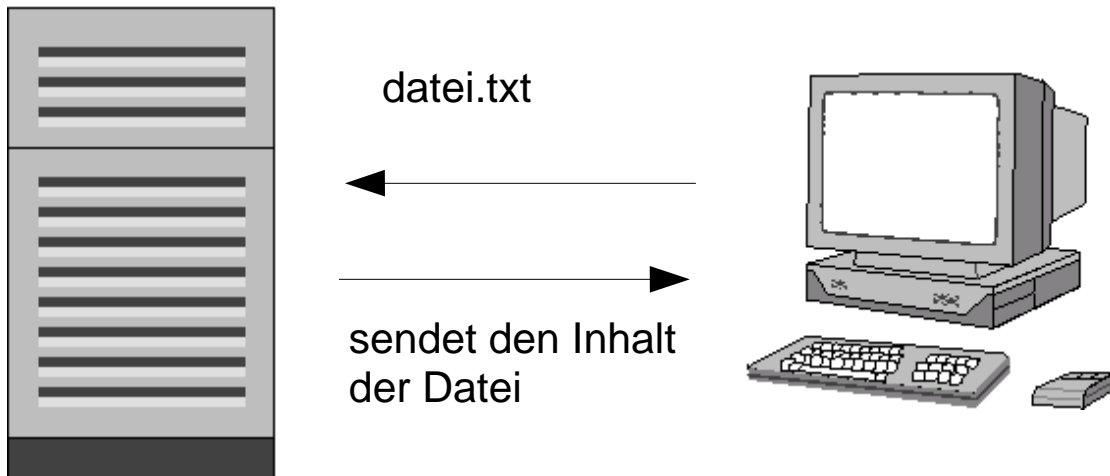
Wahl

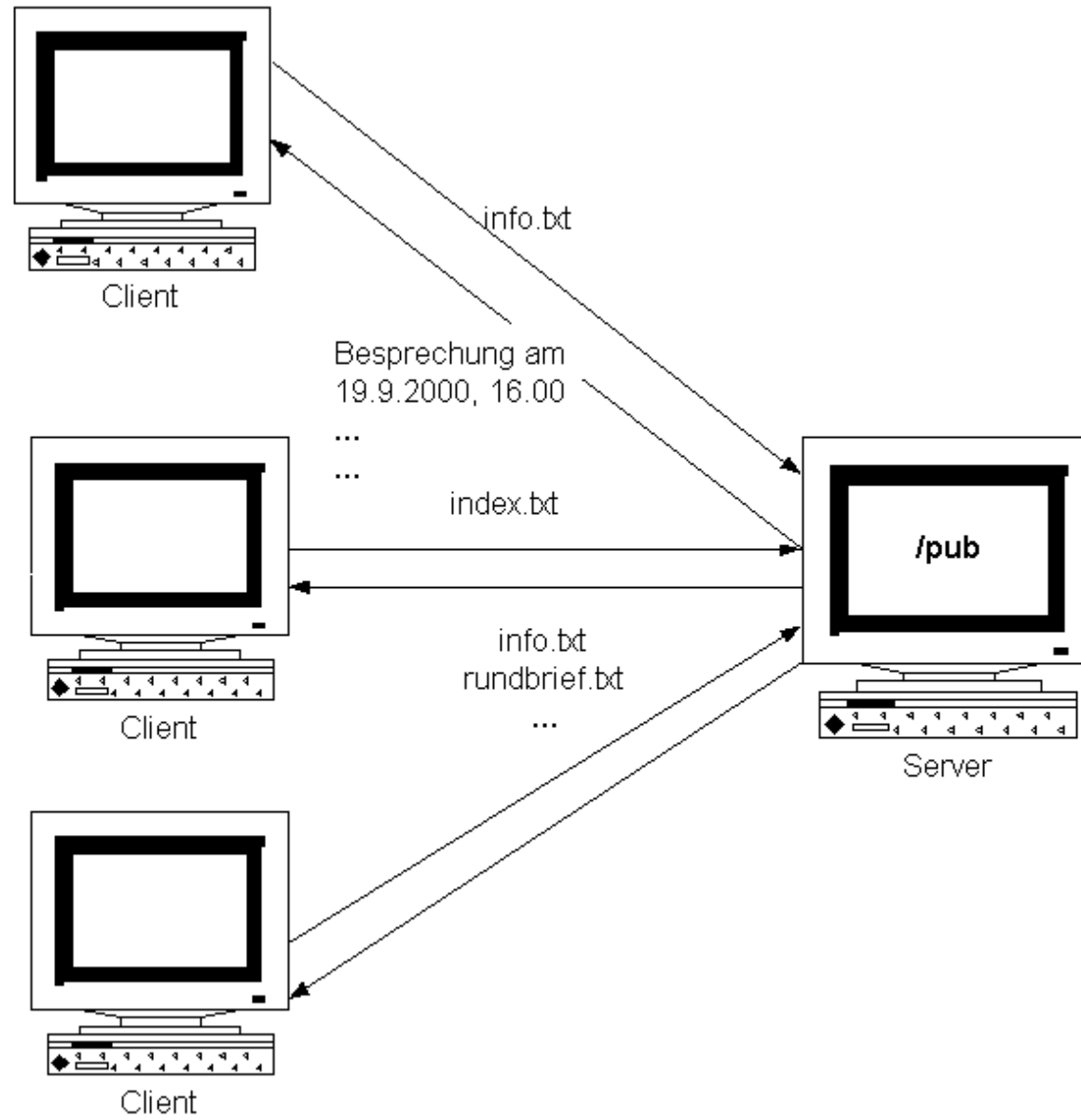


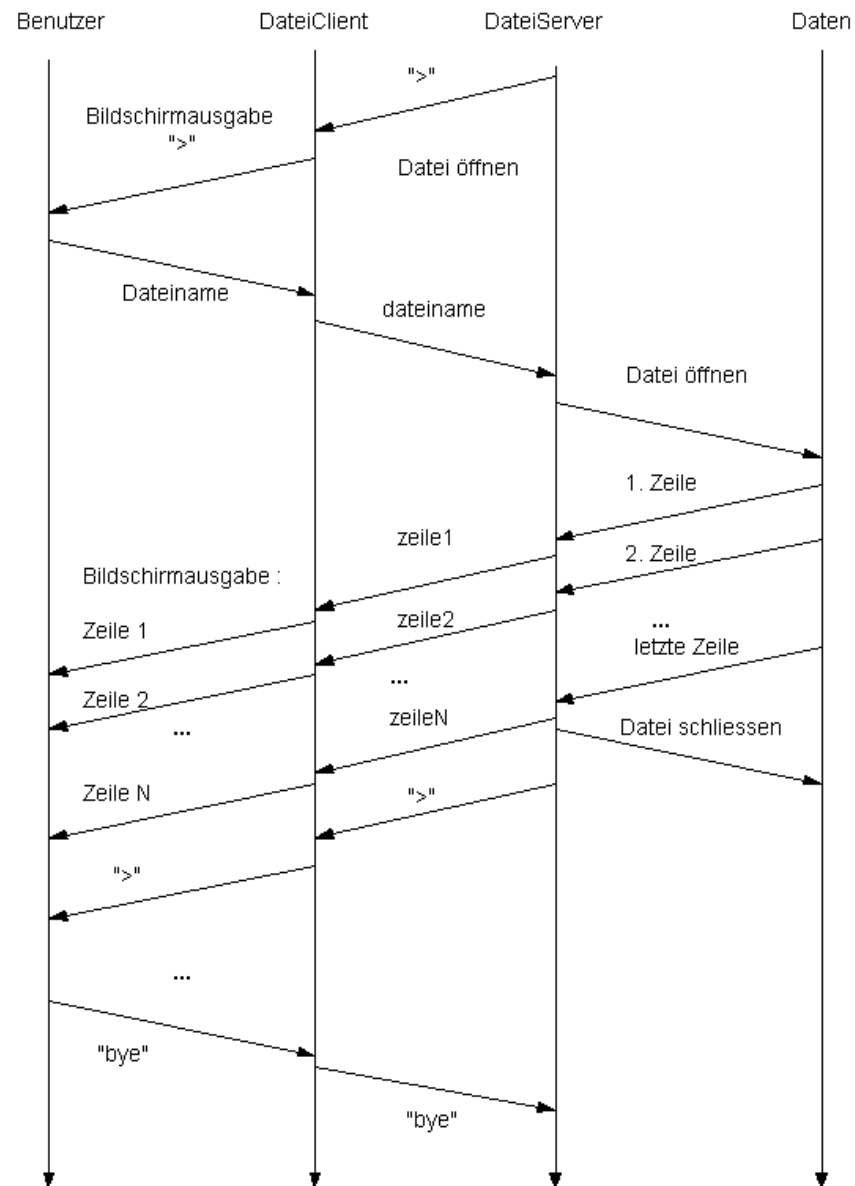
Datei-Server

Server antwortet

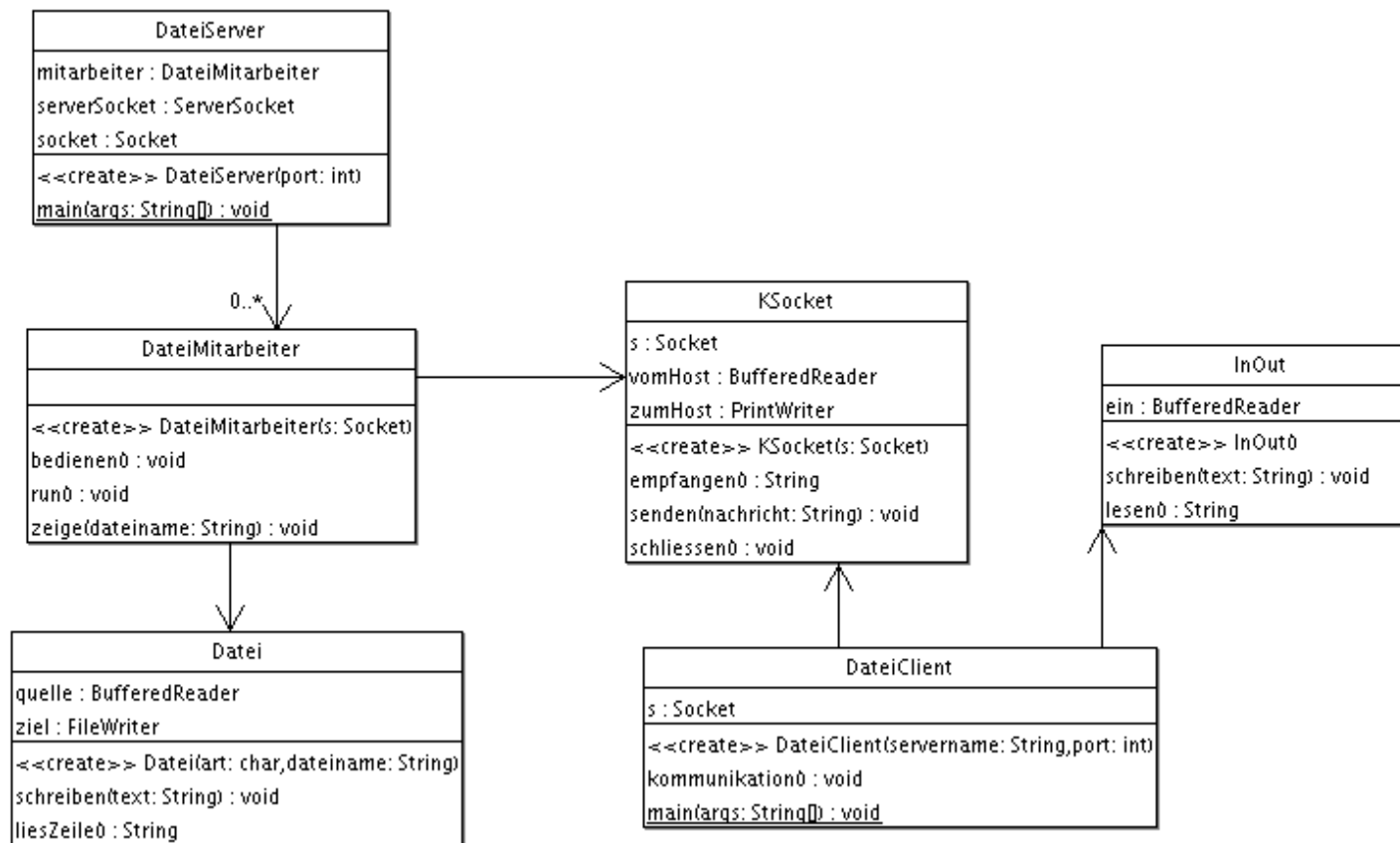
Client fragt

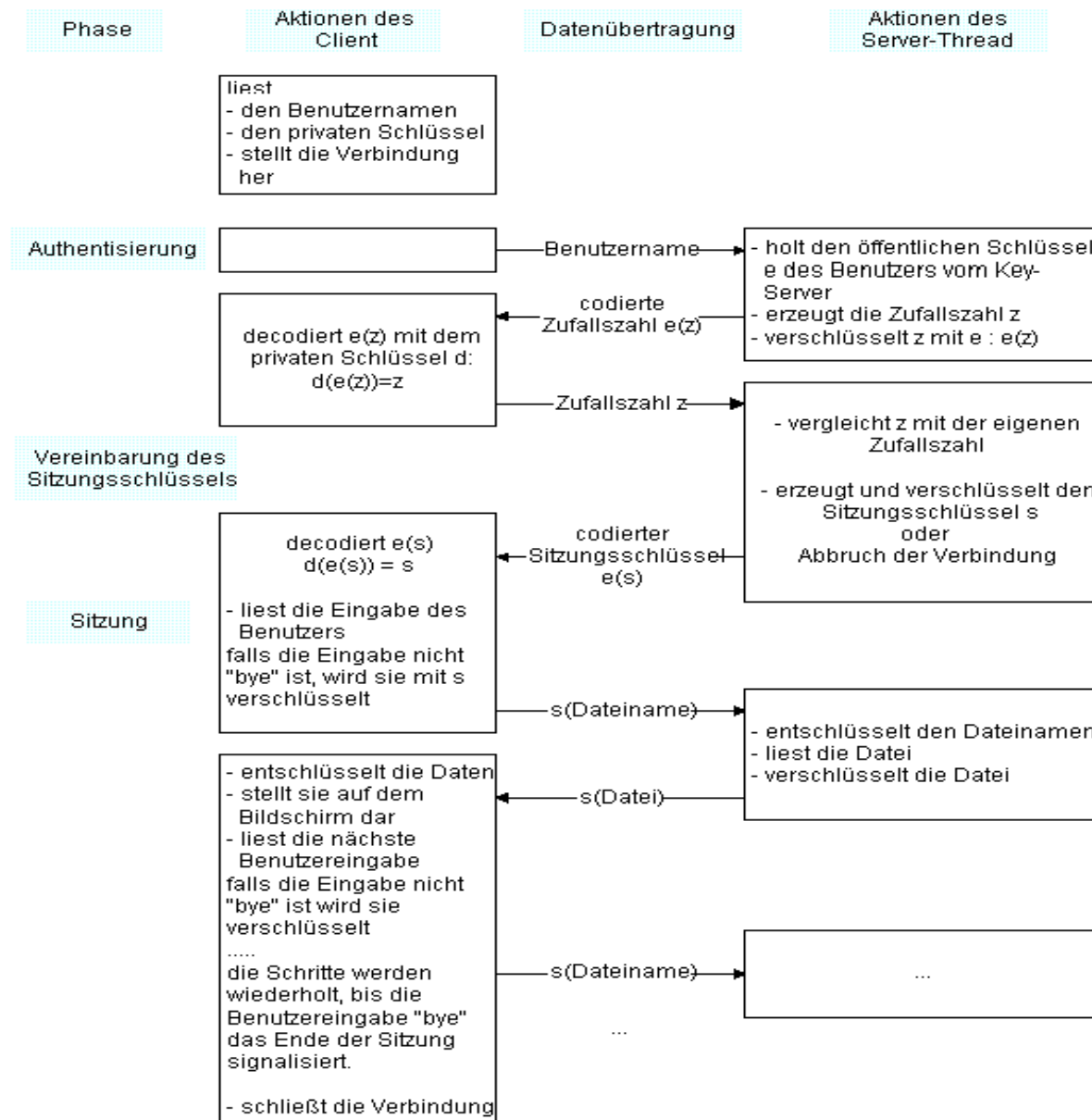




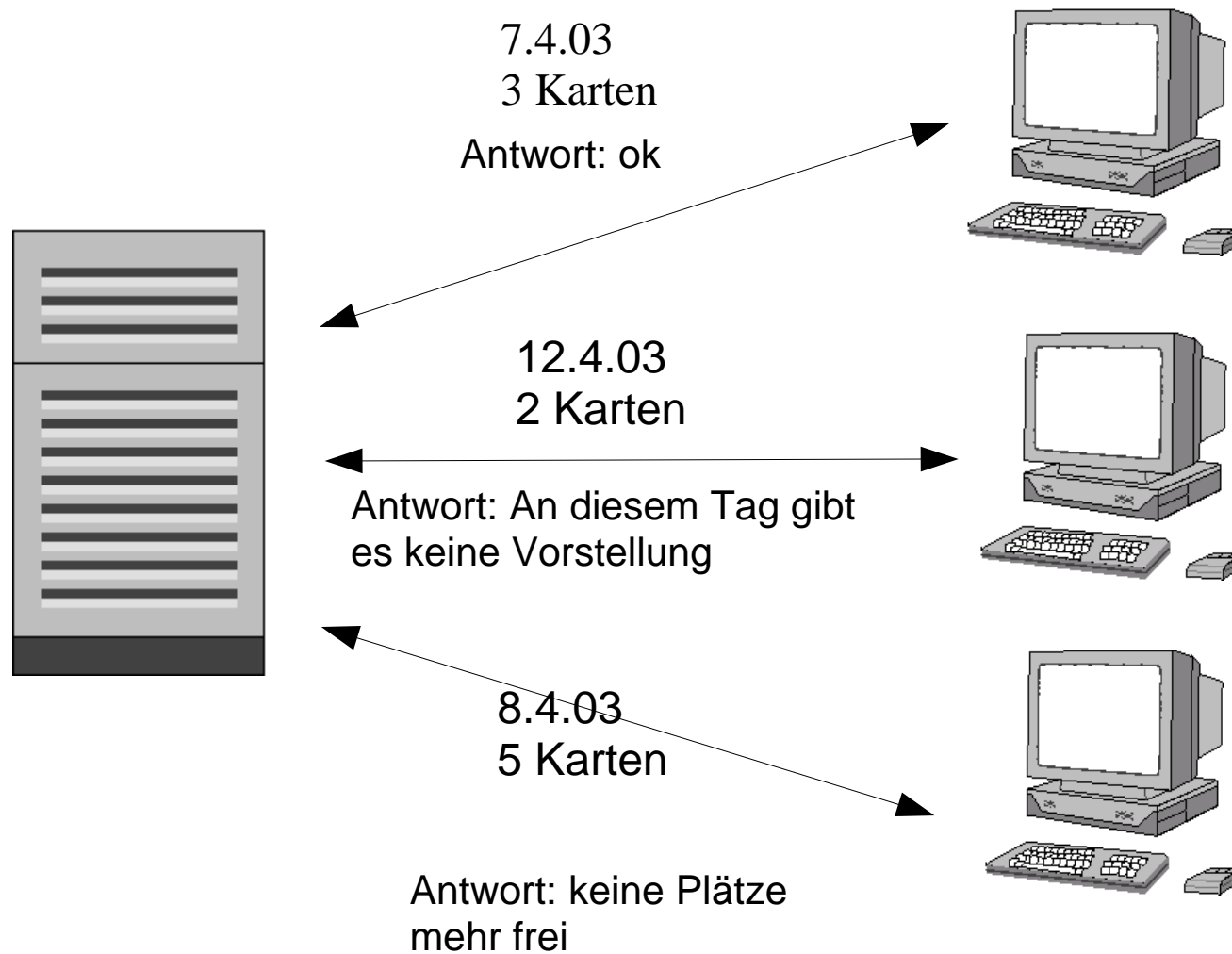


DateiServer- und -Client – UML

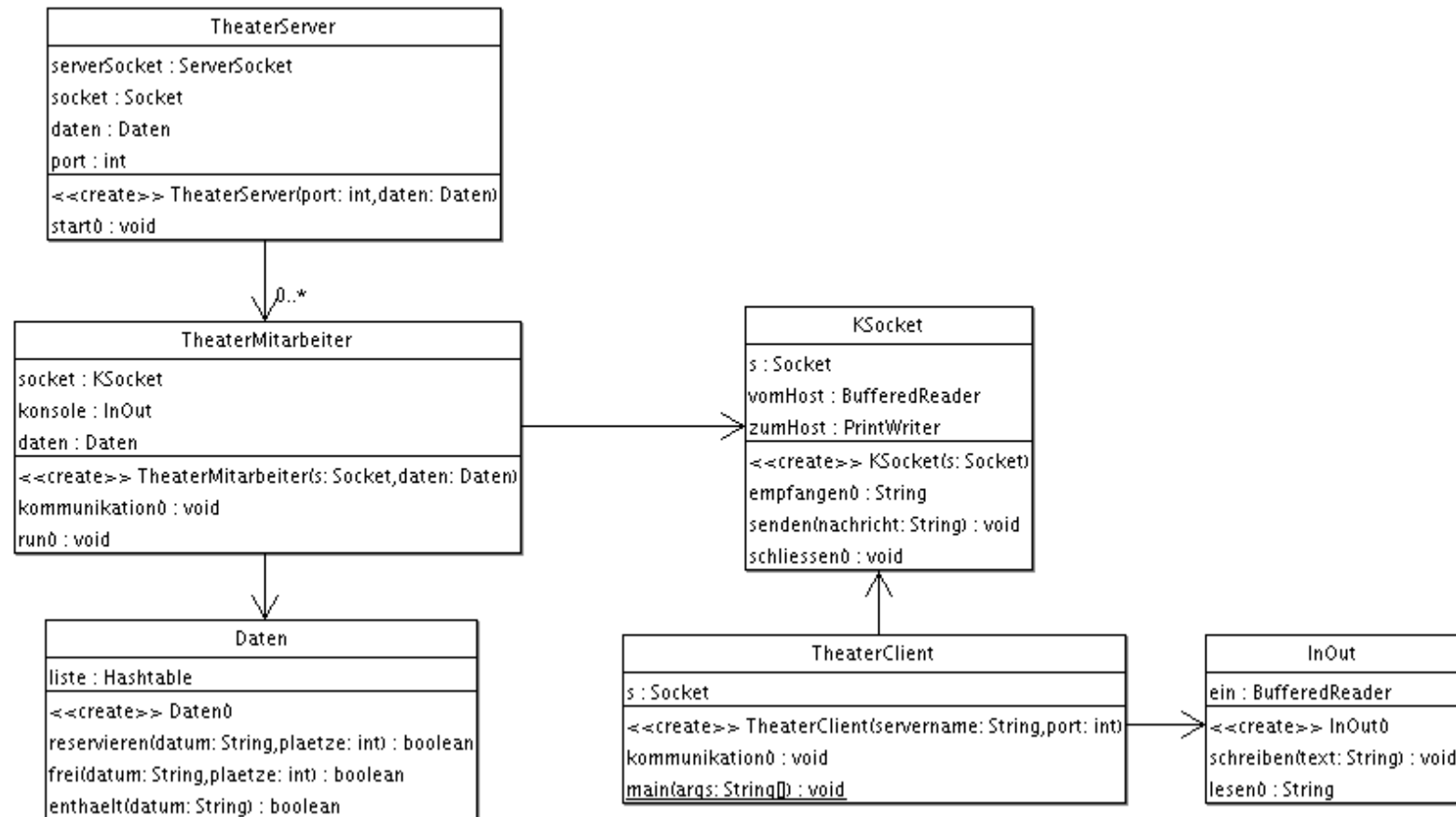




Ein Reservierungssystem



Reservierungssystem – UML



Client-Server-Applikationen mit RMI

RMI - Remote Method Invocation

Entfernter Methodenaufruf

**Client
Stub**

**Server
Skeleton**

Stellvertreterobjekt des Servers mit gleicher Schnittstelle
leitet Methodenaufrufe an den Server weiter

nimmt Aufrufe des Stubs entgegen
bereitet sie auf und übermittelt sie an das Serverobjekt
erwartet das Ergebnis und sendet es zurück an den Stub

1. RMI-Referenzschicht

Finden des Kommunikationspartners, Namensdienst (Registry)

2. RMI-Transportschicht

Verwaltung der Kommunikationsverbindung und Abwicklung der Kommunikation

3. Transportschicht

stellt Host-zu-Host-Datendienste zur Verfügung

4. Internetschicht

definiert den Aufbau von Datagrammen und routet Daten durchs Netz

5. Netzzugangsschicht

enthält Routinen für den Zugriff auf physikalische Netze

Physikalisches Übertragungsmedium

Online-Bank

1. BankServer und BankClient kompilieren

2. Stub- und Skeleton erzeugen

```
rmic BankServer -> BankServer_Stub.class, BankServer_Skel.class
```

3. Namensdienst starten

```
rmiregistry
```

4. Server starten

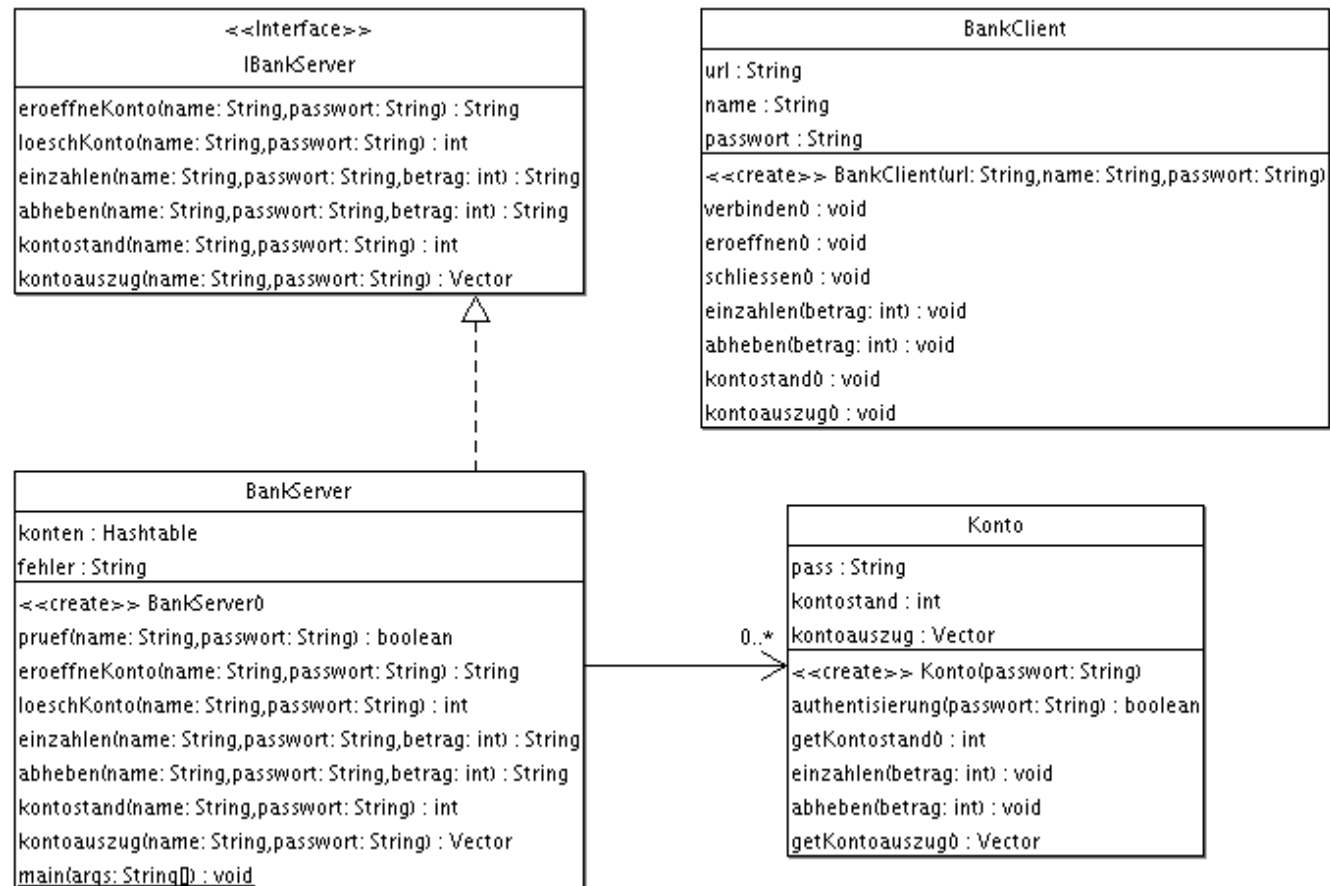
```
java BankServer
```

5. Dateien auf Client-Rechner kopieren

```
BankClient.class, BankServer_Stub.class
```

6. Client starten mit Angabe

- . Namen
- . Passwort



Chat mit RMI

