

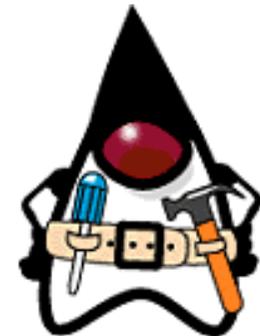
IEEE 1394-Socket

Klaus Salamonsberger

Betreuer: Dipl. Ing. Gerhard Pratl

Aufgabenstellung

- Ansprechen des Bussystems IEEE 1394 mittels Sockets
- An bestehenden Java-Socket anlehnen
- Kommunikation mit „native“ Applikationen ermöglichen
- Datenpakete nicht verfälschen



IEEE 1394

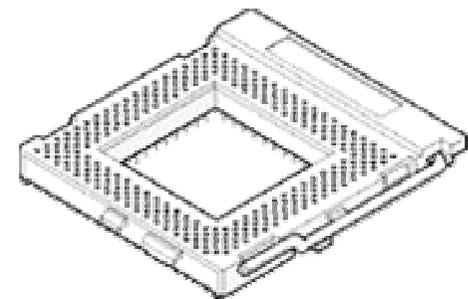


- Bekannt unter den Markennamen:
 - FireWire
 - I-Link
- Serieller Hochgeschwindigkeitsbus
 - Derzeit 400 Mbit/s
 - Spezifiziert bis 1600 Mbit/s
 - Isochroner und asynchroner Datenaustausch
 - Plug-and-play-fähig
 - Hot-Plugging
- AL 1394 (ICT)

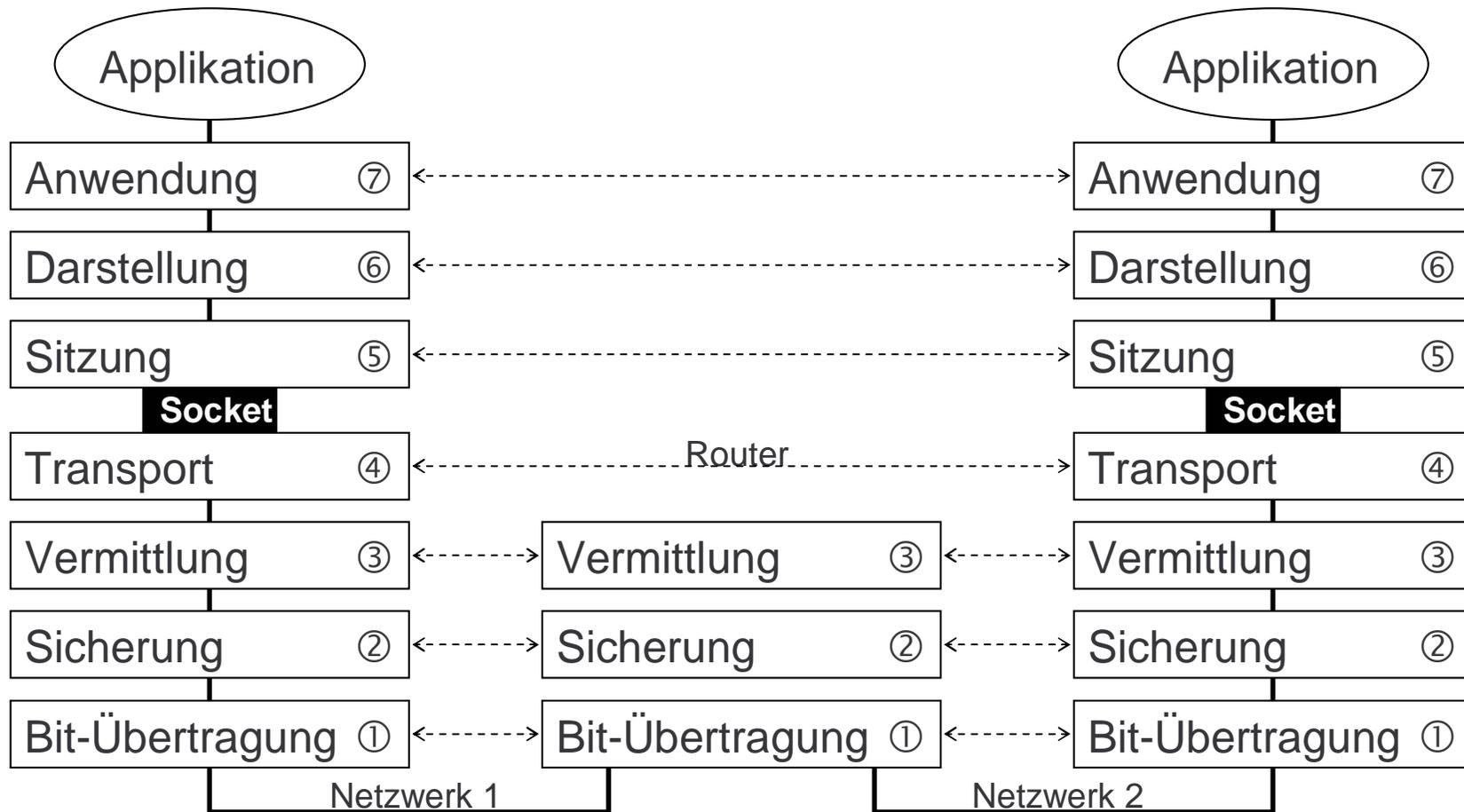


Socket

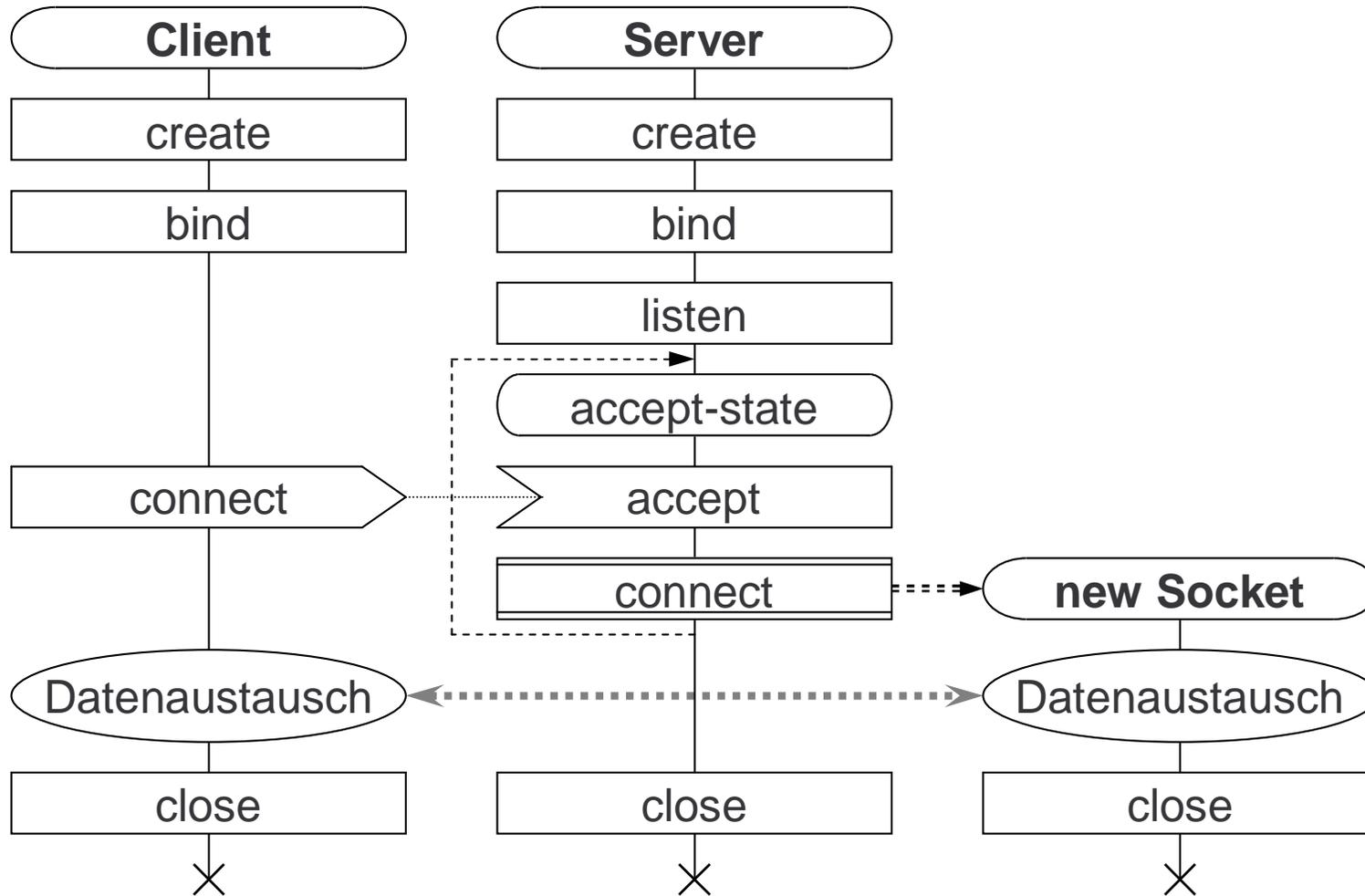
- Für Betriebssystem BSD-UNIX
(University of California at Berkley)
- Programmierschnittstelle für
Netzwerkanwendungen
- Netzwerkadresse eines Prozesses



Socket und das OSI-Referenzmodell



Verwendung von Sockets



Verwendung von Java-Sockets



Server

```
ServerSocket ss = new ServerSocket(port);  
Socket s = ss.accept ();
```

Datenaustausch

```
s.close ();  
ss.close();
```

Client

```
Socket s = new Socket (address, port);
```

Datenaustausch

```
s.close ();
```

Lesen

```
InputStream is = s.getInputStream();  
readBytes = is.read (buffer);
```

Schreiben

```
OutputStream os = s.getOutputStream();  
bytesWritten = os.write (buffer);
```

Verbindungsmanagement

- Abhängig vom Kommunikationssystem
- Verbindungsaufbau / Verbindungsabbau
- Systeme in Java
 - Socket
 - Verwendet TCP/IP
 - ServerSocket
 - File



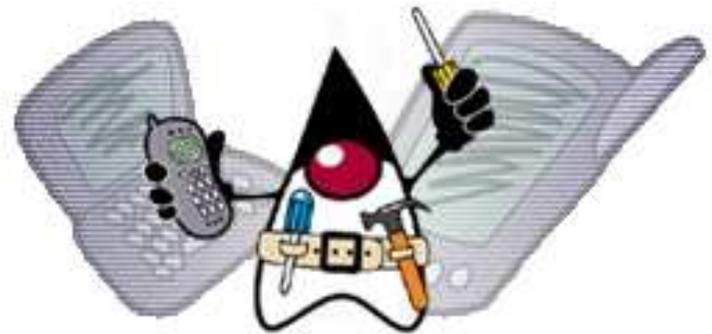
Datenaustausch

- Medium ist unterschiedlich
- Methodik für alle Kommunikationssysteme gleich
 - Read
 - Write
 - (Lock)
- In Java
 - InputStream
 - OutputStream



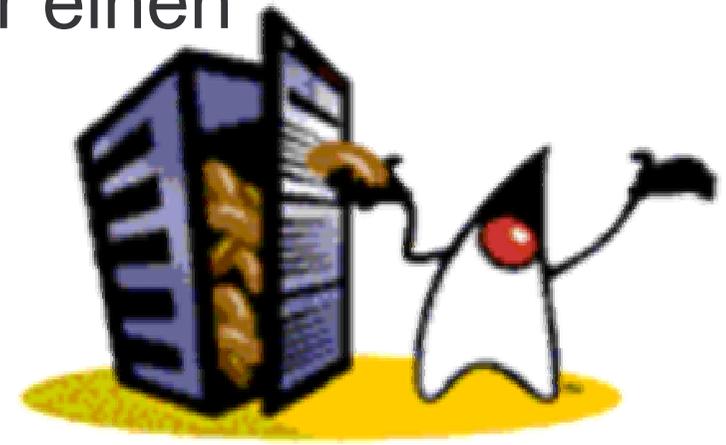
Lösungsansätze

- Abbilden der gesamten IEEE 1394-API
- Eigene Protokollschicht implementieren
- Teile der IEEE 1394-API abbilden
- Teile der IEEE 1394-API abbilden und mit einem Kontrollmechanismus erweitern



Gewählter Lösungsansatz

- Socket vollständig in C/C++ implementieren
 - Java verwendet mittels JNI diesen Socket
- Zwei Kommunikationsarten:
 - Kommunikation mit einem IEEE 1394-Socket
 - Kommunikation mit einer IEEE 1394-Applikation
- Verbindungsmanagement über einen zusätzlichen Kanal
- Zwei Socket-Typen:
 - Isochron
 - Asynchron



Schnittstellen

- IEEE 1394-Socket in Java
 - Vorgegeben durch Aufgabenstellung
- JNI (Java Native Interface)
 - Datenaustausch zwischen Java und anderen Sprache
- IEEE 1394-Socket in DLL
 - Implementierung in C/C++
 - Ist bereits ein voll funktionstüchtiger Socket
- AL 1394 (entwickelt am ICT)
 - Ermöglicht den Zugriff auf IEEE 1394-Bus

