



Socketprogrammierung

Thomas Karp

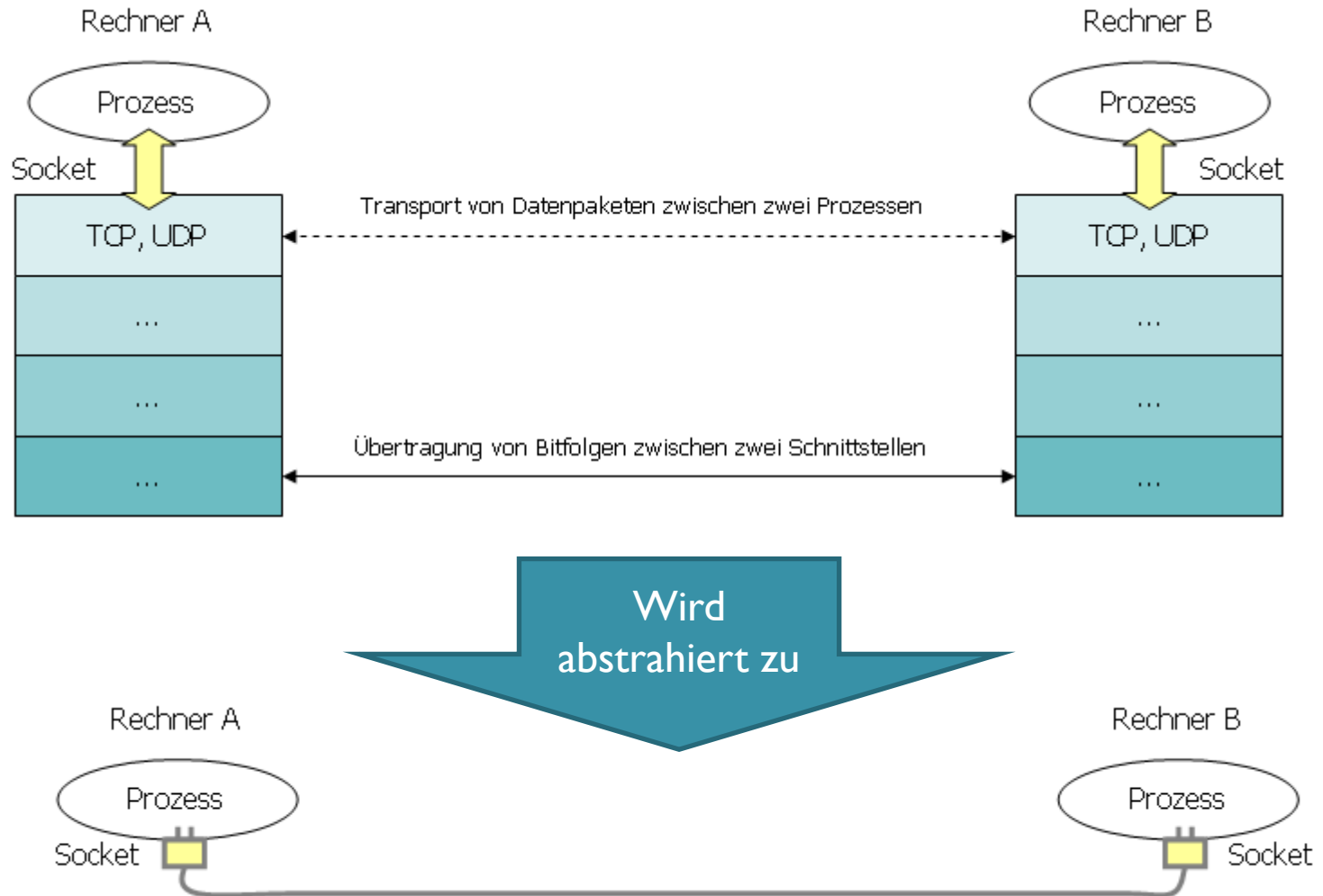
—

Friedrich-Magnus-Schwerd Gymnasium Speyer

Unterrichtsideen im Netz

- Infoskript.de
 - Java als Programmiersprache
 - Kapitel zur Socketprogrammierung knapp aber vollständig
 - Wird evtl. teilweise auf inf-schule.de integriert
 - Kapitel: Rechnernetze → Sockets
- Inf-schule.de
 - Python als Programmiersprache
 - Kapitel zur Socketprogrammierung umfangreicher, aber zur Zeit noch mit Lücken
 - Kapitel: Kommunikation in Rechnernetzen → Internetanwendungen selbst programmieren

Was ist ein Socket?



Vergleich mit Telefonverbindung

- Rufnummer (entspricht IP-Nummer)
- Durchwahl (entspricht Port-Nummer)
- Anrufer wählt (entspricht Verbindungsaufbau durch Client)
- Angerufener hebt ab (entspricht Verbindungsannahme durch Server)
- Danach: Verbindung symmetrisch und fullduplex
- Gespräch je nach Protokoll
- Beendigung durch Auflegen
- Eigene Nummer muss nicht bekannt sein

Aus RFC – Request for comments

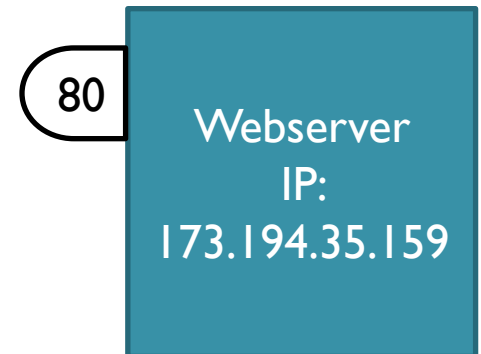
- Socket ist Tupel aus Ziel- und Quell-IP-Adresse, Ziel- und Quell-Port (nach Wikipedia)
- IP:Adresse des Rechners, z.B.
95.88.106.176
- Port-Nummer: Reservierte Nummer des Anwendungsprozesses, z.B.
 - 35788 als zufällig gewählter Port von Firefox
 - 80 als Port des Webserver

Well known ports

- Portnummer ist 16Bit-Zahl, Bereich also von 0 bis 65535
- Well known ports sind Ports, unter denen eine Server üblicherweise erreichbar ist
 - 53 für DNS (Domain Name System)
 - 80 für HTTP (Webserver)
 - 110 für POP3 (Email-Abruf)
- Port 0 bis 1023 sind standardisiert
 - → Well known ports
- Port 1023 bis 49151 sind registriert
- Ports ab 49152 zur freien Verfügung

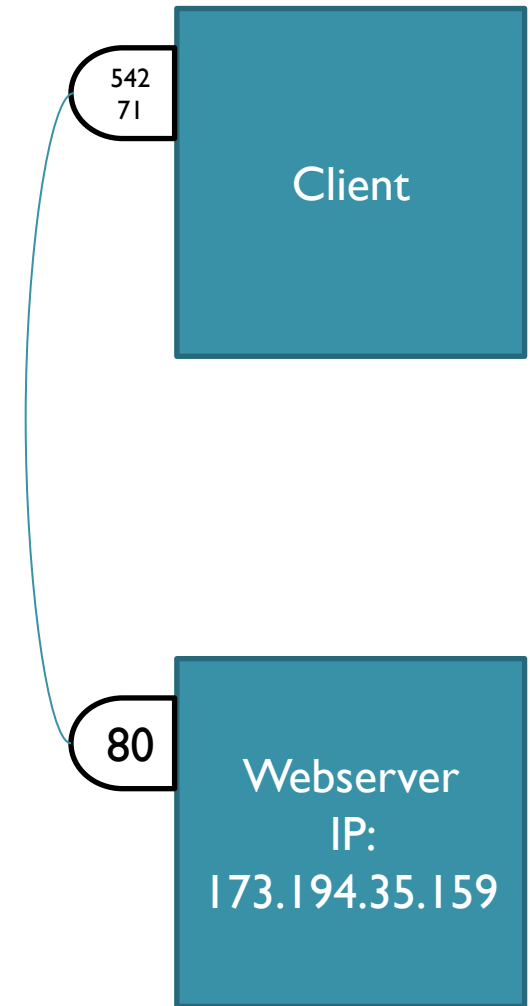
Verbindungsaufbau

- Webserver wartet auf Port 80



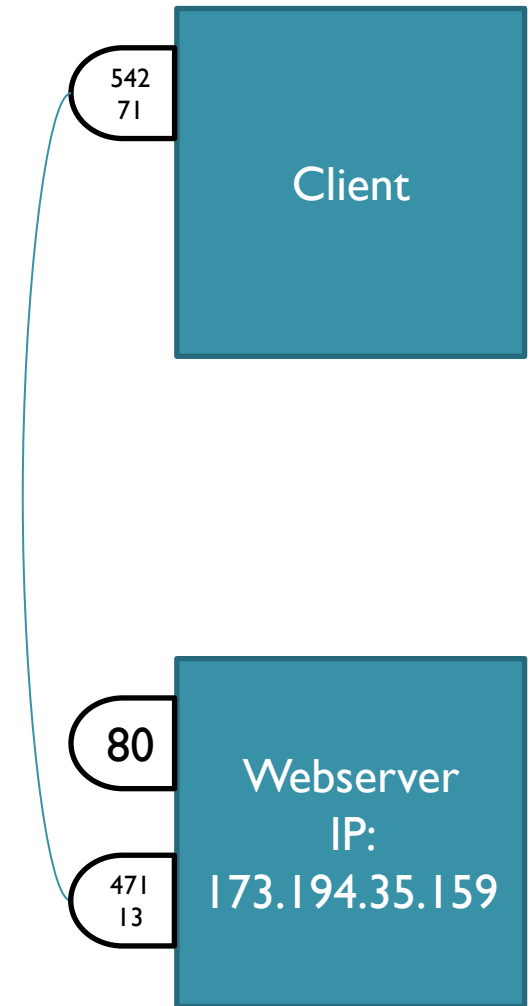
Verbindungsaufbau

- Webserver wartet auf Port 80
- Client verbindet sich

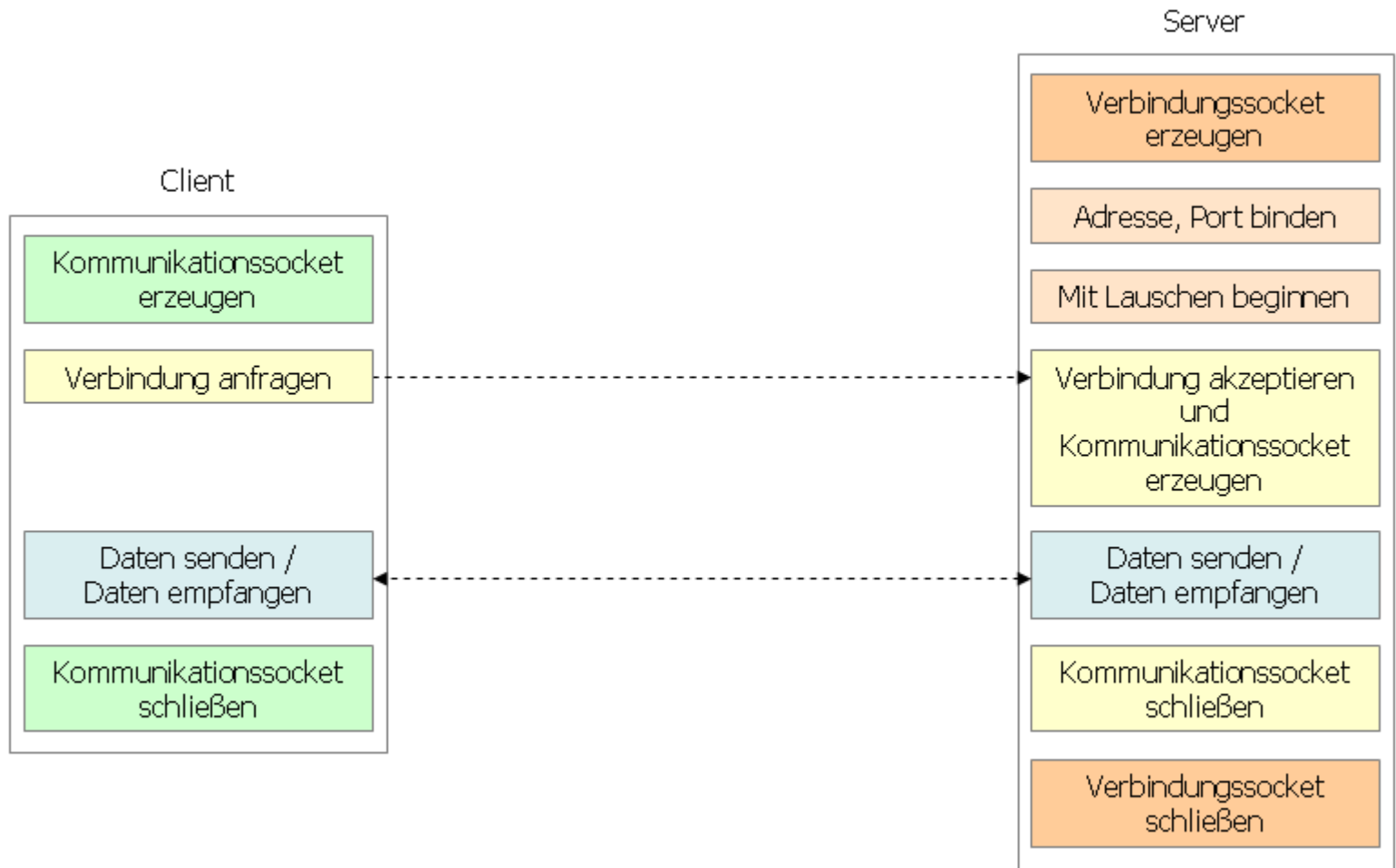


Verbindungsaufbau

- Webserver wartet auf Port 80
- Client verbindet sich
- Server erzeugt neues Socket auf freiem Port und stellt Verbindung darüber bereit
- Port 80 ist wieder frei für weiteren Client



Client-Server-Kommunikation



Nach inf-schule.de

Mögliche Aufgaben

- Python
 - Ein HTTP-Client ([Kapitel 1.16.3.2.3](#))
 - Ein Echo-Server ([Kapitel 1.16.3.2.4](#))
- Java
 - [Sequenz](#) auf infoskript.de
 - [Witzeserver](#) mit pdf-Hilfe, Testen mit Universalclient