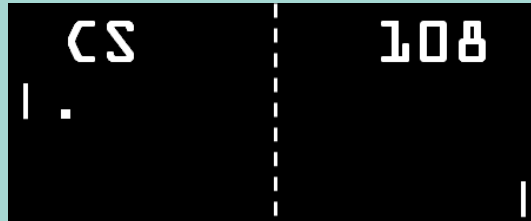


Kapitel 3 – Client/Server-Architekturen

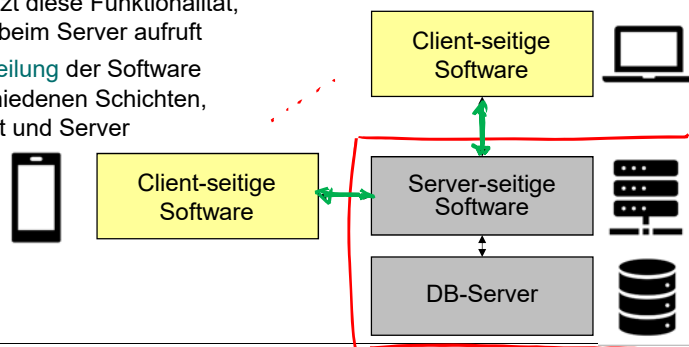
cs108 Programmierprojekt

H. Schuldt, FS 2020



Client/Server-Systeme

- Ein zweischichtiges Client/Server-System ist die einfachste Variante eines **verteilten Systems** (also eines Systems, in dem die Komponenten über Rechnergrenzen hinweg verteilt und über ein Netzwerk verbunden sind)
- **Spezielle** Dienste, die für mehrere Nutzer relevant sind, werden an einer zentralen Stelle beim **Server** bereitgestellt
- Der **Client** nutzt diese Funktionalität, in dem er sie beim Server aufruft
- **Vertikale Verteilung** der Software auf die verschiedenen Schichten, also auf Client und Server



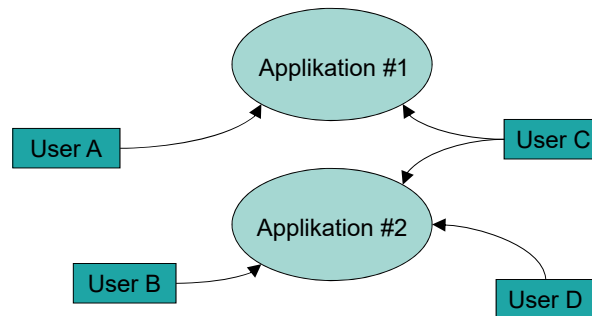
FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-2

Netzwerk-Anwendungen

Ziel: Nutzung gemeinsamer Ressourcen (Daten/Dienste) in einem Netzwerk, z.B.:

- mehrere Anwender müssen auf eine Datei zugreifen
- mehrere Arbeitsplätze sollen auf einen gemeinsamen Drucker zugreifen
- mehrere Programme möchten eine gemeinsame Anwendungsfunktionalität (Anwendungsdienst) nutzen, z.B. Mail-Server, Web-Server
- ...



FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3- 3

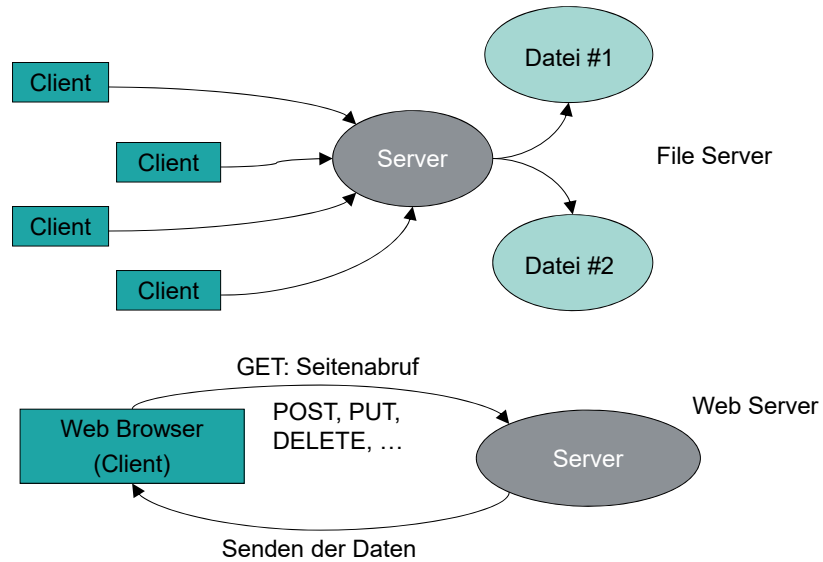
Client/Server-Modell

- Die Kommunikation zwischen zwei Rechnern läuft oft auf der Basis einer Client-Server Beziehung ab
- **Server**
 - Der Server stellt einen Dienst zur Verfügung, der von anderen Rechnern genutzt werden kann. Dieser Dienst läuft im Hintergrund und wartet darauf, dass ein anderer Rechner (Client) eine Verbindung aufbaut
 - Der Server definiert das Protokoll, mit dessen Hilfe der Datenaustausch erfolgt
- **Client**
 - Nutzt den Dienst eines oder mehrerer Server und baut bei Bedarf die Client/Server Verbindung entsprechend dem Server-Protokoll auf

FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3- 4

Beispiele für Client/Server-Anwendungen ...



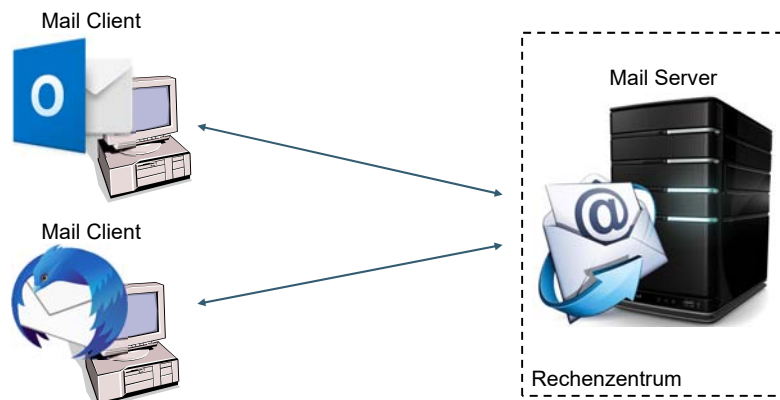
FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-5

... Beispiele für Client/Server-Anwendungen ...

Mail Server

- Mail-Programme (Clients) kommunizieren über ein Netzwerk mit einem Mail Server, der in der Regel in einem physisch getrenntem Raum steht.



FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-6

Server/Client-Protokoll

- Um Daten zwischen zwei oder mehreren Partnern austauschen zu können, müssen sich die Teilnehmer auf ein **Protokoll** geeinigt haben
- Als **Protokoll** bezeichnet man die Menge aller Regeln, die notwendig sind, um einen eindeutigen **Verbindungsaufbau**, **Datenaustausch** und **Verbindungsabbau** zu gewährleisten
- Das Protokoll ist die entscheidende Komponente einer Client-Server Verbindung
- Fundamentale Anforderungen:
 - **eindeutige Definition** (ein Befehl kann nicht unterschiedlich interpretiert werden)
 - **vollständig** (alle Situationen sind klar definiert, für jede Funktion existiert ein Befehl)
 - **erweiterbar** (zukünftige Änderungen sind möglich)
 - **gut dokumentiert**

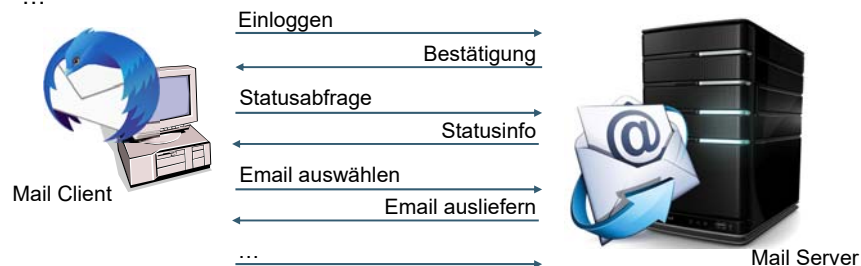
FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schultt 3-7

... Beispiele für Client/Server-Anwendungen

Kommunikation zwischen Mail Client (C) und Mail Server (S)

- Einloggen (C → S)
- Bestätigung (S → C)
- Statusabfrage (C → S)
- Statusinformation (S → C)
- Email auswählen (C → S)
- Email ausliefern (S → C)
- ...



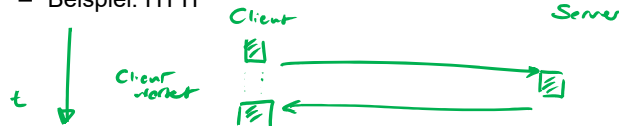
FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schultt 3-8

Protokoll-Arten

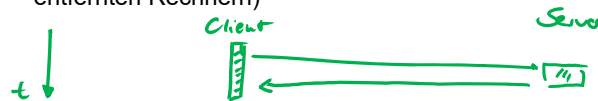
- **Synchron:** ein Client sendet eine Abfrage, der Server antwortet sofort, in der Zwischenzeit wartet der Client auf die Antwort des Servers (übernimmt keine andere Aktivität).

– Beispiel: HTTP



- **Asynchron:** Client und Server kommunizieren gleichzeitig. Der Client richtet eine Anfrage an der Server. Während der Server diese bearbeitet kann der Client mit seiner Arbeit fortfahren, er muss nicht auf das Ergebnis des Servers warten.

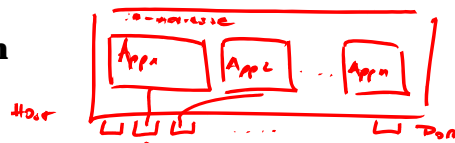
– Beispiel: TELNET, RLOGIN (Arbeiten auf anderen z.B. physikalisch entfernten Rechnern)



FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-9

Ports und Applikationen



- Auf einem Host laufen meist unterschiedliche Serveranwendungen, die von mehreren Clients gleichzeitig benutzt werden
- Um die Server voneinander zu unterscheiden, gibt es ein weiteres Adressmerkmal, die **Port-Nummer**
- Diese gibt die Serveranwendung an, mit der ein Client kommunizieren will
- Port-Nummern sind positive ganze Zahlen im Bereich von 0 bis 65535 ($2^{16}-1$) (0 bis 1023 sind reserviert)
- Jeder Servertyp hat seine eigene Port-Nummer, viele sind bereits für sehr weit verbreitete Client/Server-Anwendungen reserviert:

Name	Port-Nummer	Beschreibung
echo	7	gibt den Text zurück, den der Client sendet
pop3	110	Übertragen von Mails
www	80	Web Server
...

FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-10

Beispiel: POP3 ...

- POP3 (Post Office Protocol) wird benutzt, um Mails vom Server auf den Client zu übertragen. Es ist ein textbasiertes, synchrones Protokoll.
- Protokoll-Definition: RFC1939 (Request for Comments)

Beispiel-Session (synchron):

```

limmat.cs.unibas.ch/home> telnet ip-Adressepop.gmx.net Port110
Trying 213.165.64.20...
Connected to pop.gmx.net.
Escape character is '^]'.
+OK GMX POP3 StreamProxy ready
<17613.1110544480@mp013>
  
```

FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-11

... Beispiel: POP3 ...

- Einloggen mit Username und Passwort: ^{Protokollbefehl}

```

USER testname@gmx.net      → Client → Server  C → S
+OK May I have your password, please?  S → C
PASS *****              C → S
  
```
- Status der Mailbox überprüfen:


```

STAT                        C → S
+OK 2 16914                 S → C
      
```
- Eine Nachricht lesen:


```

RETR 1                      C → S
+OK message follows         S → C
<message headers and body>
      
```

FS 2020

Programmierprojekt (cs108) – Client/Server-Architekturen – Heiko Schuldt 3-12

... Beispiel: POP3

- Eine Email löschen und die Verbindung beenden:

```
DELE 1
+OK message deleted
QUIT
+OK bye
Connection closed by foreign host.
limmat.cs.unibas.ch/home>
```

Diskussion

- Anfragen erfolgen Text-basiert (z.B. **RETR 1**)
- Antwort besteht aus einer Status-Meldung (z.B. **+OK**) und der Mail.

Wo ist dies in unserem Projekt relevant?

Aus den Projektanforderungen:

„Der Server und die verschiedenen Clients kommunizieren mit Hilfe eines selbst erstellten textbasierten Netzwerk-Protokolls. Das Protokoll muss lesbar sein, um eine Fehlersuche zu vereinfachen.“

POP3 auf der Kommandozeile	Ihr Terminal (Meilenstein 2) bzw. Ihr GUI (Meilenstein 3 und danach)
Textbasiert	Textbasiert ✓
Lesbar	Lesbar ✓
Synchron / interaktiv (login)	? Asynchron
Eingabe muss exakt gemäss POP3-Protokoll erfolgen	Eingabe der zu übertragenden Befehle soll unabhängig vom Netzwerk-Protokoll sein (siehe Malus «Human Modem»)

Details zur Netzwerkprogrammierung in Java folgen in Kapitel 5.