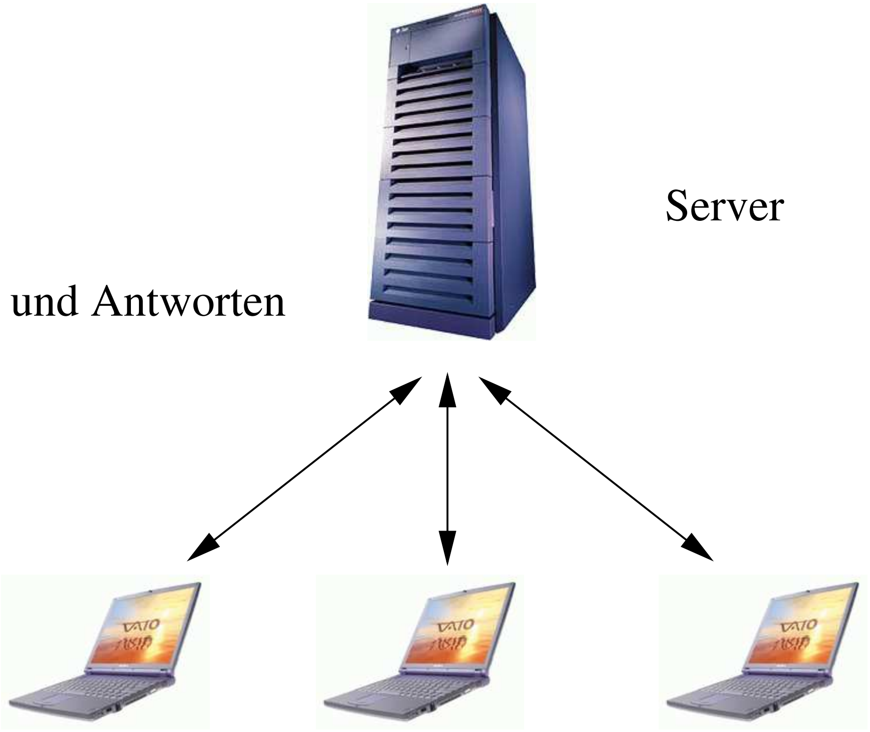
Praktikum: Entwicklung eines Lehrveranstaltungssystem



Daten pro Lehrveranstaltung

Veranstaltungskennziffer: VAK — eindeutig *03-763*

Titel *Internet-Programmierung*

Veranstalter: Name und Vorname *Christoph Lüth, Hui Shi*

Termin: Ort und Zeit *Di 08-10 MZH 7260, Do 08-10 MZH 7260*

Inhalt der Veranstaltung als Liste von Themen
Einführung, Nebenläufigkeit und Synchronisation, . . .

Diskussion als Listen von Fragen und Antworten

Benutzerführung des Clients

- Hauptfenster:

  Auswahlmöglichkeiten (Knopf/Menü) für Suche und Programmende

  Übersichtsanzeige aller ausgewählten Kurse

  Detailansicht (alle Daten und Diskussion) für einen ausgewählten Kurs

- Suchmaske:

  Eingabefelder für VAK, Titel, Veranstalter, Ort, Zeit

Datenaustausch zwischen Client und Server

- Format: XML
- Server an Client: Kursdaten (für einen oder mehrere Kurse)
  `kurs.dtd`, Daten für einen Kurs: Element `kurs`
- Client an Server: `anfrag.dtd`
  - Anfrage (Element `anfrag`)
  - Neue Frage (Element `neuefrg`)
  - Neue Antwort (Element `neueantw`)
  - Konvention: erste gesendete Zeile ist der Benutzername
- Datenaustausch nebenläufig:
  - Client kann jederzeit neue Daten und Antworten senden.
  - Server kann jederzeit neue Daten senden.
  - Insbesondere propagiert der Server Änderungen an *interessierte* Clients.

Bewertungskriterien

- Aufgabenzettel definiert *Grundanforderungen*, Erweiterungen möglich und erwünscht
- Implementierungssprachen: Java oder Haskell
- Wichtig für Bewertung:
  - Nebenläufigkeit und Synchronisation bei neuen Fragen und Antworten
  - Modularisierung: Server und Client sollten unabhängig sein, jeder Server sollte mit jedem Client arbeiten
  - Stabilität des Servers: alle Fehlerfälle des Clients abfangen
- Nicht so wichtig: Details des GUI