

— VS —

**Konzept**

# Lernziele

- zu erkennen, was über die „bloße“ Vernetzung von Rechnern hinausgeht
  - was z.B. Rechnernetze sind, ist nicht Thema der Veranstaltung
- zu erfahren, welchen *Schwerpunkt Middleware* dabei besitzt
  - Verteilte Systeme aus Sicht der Systementwicklung beleuchten
- zu ermöglichen, ein Fernaufrufsystem von der Pike auf zu entwickeln

# Lehrinhalte (1)

**Eigenschaften** ..... Heterogenität, Offenheit, Sicherheit, Skalierbarkeit, Fehlerverarbeitung, Nebenläufigkeit, Transparenz ..... (1)

**Kommunikationssystem** ..... Nachrichtenaustausch, Rendezvous; Pufferung; synchrone vs. asynchrone Kommunikation ..... (1)

**Fernaufruf** ..... Semantikaspekte; zusammenstellen („*marshalling*“) und auseinandernehmen („*unmarshalling*“) von Botschaften, Repräsentation von Daten; Zustellungsgarantien, Idempotenz; Rückrufe; asynchroner – ... (2)

**Lokalisierung** ..... Namen, Adressen und Attribute; Namensraum, Namensauflösung; Namens-, Verzeichnis- und Erkennungsdienst ..... (2)

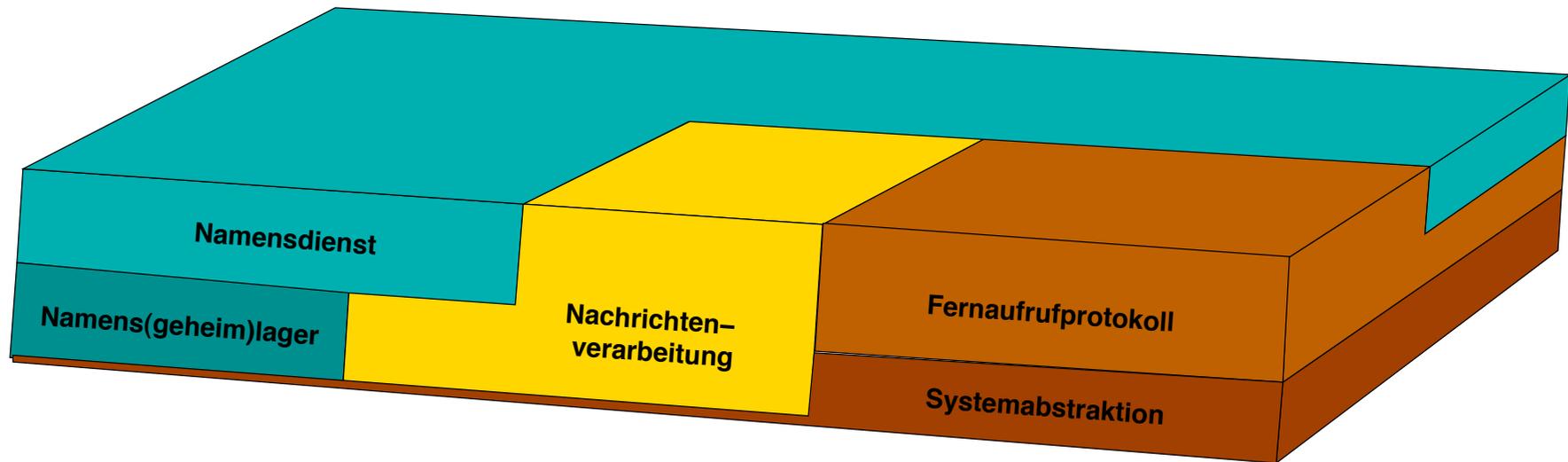
## Lehrinhalte (2)

**Netzwerktransparenz** ..... linguistische Unterstützung, „Stümpfe“;  
Objektorientierung, Versprechen ..... (2)

**Replikationstransparenz** ..... verteilte Dateisysteme (NFS, AFS),  
verteilter gemeinsamer Speicher (Ivy, Munin, Vote); „Geheimlagerung“, Kon-  
sistenzmodelle, Aktualisierungsoptionen ..... (2)

**Systemmodelle** ..... Architekturkonzepte (Client/Server,  
Proxy-Server, schlanke Klienten, gleichrangige Prozesse; mobiler Kode, mobile  
Agenten, mobile Geräte); Entwurfsanforderungen; Interaktions-, Fehler- und  
Sicherheitsmodell ..... (2)

# Übungsstoff (1)



**FAX** Eine plattformunabhängige, portable, „federgewichtige“ Fernaufrufexekutive für heterogene, vernetzte Rechnersysteme.

## Übungsstoff (2)

1. *Systemabstraktion*: Abstraktion von der Schnittstelle und den Eigenschaften der jeweils zu Grunde liegenden Systemplattform.
2. *Nachrichtenverarbeitung*: Elementare Operationen zur Verarbeitung problemorientierter Nachrichtenformate.
3. *Fernaufrufprotokoll*: Prozedurfernaufrufe ohne und mit Zustellungsgarantien.
4. *Namensdienst*: Lokalisierung dienstbringender Instanzen bzw. Prozesse.
5. *Namens(geheim)lager*: Beschleunigung des Lokalisierungsvorgangs.

Laufzeitmessung und -bewertung auf Basis von *Benchmarks* für alle Aufgaben.

# Einordnung

- Studiengänge
  - Informatik, Hauptstudium (s.u.)
  - Wirtschaftsinformatik (Wahl)
  - Computational Engineering (Bachelor/Master) (Wahl)
- *Wahlpflicht* für „Verteilte Systeme und Betriebssysteme“ . . .
  - als  $\left\{ \begin{array}{l} \text{Prüfungsfach} \quad (8 \text{ SWS}) \\ \text{Schwerpunktfach} \quad (12 \text{ SWS}) \end{array} \right\}$  *Prüfung/Schein* ( {,un}benotet)
- sehr empfohlen in Kombination mit **Verteilte Algorithmen** oder **Middleware**

# Voraussetzungen

- Systemprogrammierung (Betriebssysteme), Rechnernetze
- C/C++
- sich an systemnaher Programmierung erfreuen können
  - „Furchtlosigkeit“ vor nur schwer erkund- und fassbaren Sachverhalten
- ein gewisses Maß an Durchhaltevermögen

# Organisation

- **integrierte Lehrveranstaltung**,  $2 \times 1,5$  Stunden wöchentlich.....4 SWS  
VL Vorstellung und detaillierte Behandlung des Lehrstoffs  
Ü Vertiefung, Besprechung der Übungsaufgaben, Tafelübungen
- **Rechnerarbeit**,  $M$  Stunden wöchentlich  $0 < M \leq 165$  ..... 0 SWS  
– am Rechner unbetreutes Bearbeiten der Übungsaufgaben
- **Vor-/Nacharbeit**,  $N$  Stunden wöchentlich,  $0 \leq N \leq (165 - M)$  ..... 0 SWS

# Leistungskontrolle

- **Schein**
  - *Ausstellung* ..... bei erfolgreicher Bearbeitung aller Aufgaben
  - *Rücksprache* ..... bei nicht-erfolgreicher Bearbeitung einer Aufgabe
- **benoteter Schein** ..... *Voraussetzung*: Schein (s.o.)
  - abschließendes Gespräch („Scheinprüfung“) über den Übungsstoff
- **Prüfung** bei Wahl für „Verteilte Systeme und Betriebssysteme“
  - Inhalt ist Vorlesungs- und Übungsstoff (d.h. die „Scheinaufgaben“)

# Lehrkörper

- Wolfgang Schröder-Preikschat ..... Professor
  - <http://www4.informatik.uni-erlangen.de/~wosch>
- Meik Felser ..... Mitarbeiter
  - <http://www4.informatik.uni-erlangen.de/~felser>
- Rüdiger Kapitza ..... Mitarbeiter
  - <http://www4.informatik.uni-erlangen.de/~rrkapitz>

## Empfohlene Literatur

- [1] G. Coulouris, J. Dollimore, and T. Kimberg. *Verteilte Systeme: Konzepte und Design*. Pearson Education, 2002. ISBN 3-8273-7022-1.
- [2] R. G. Herrtwich and G. Hommel. *Kooperation und Konkurrenz — Nebenläufige, verteilte und echtzeitabhängige Programmsysteme*. Springer-Verlag, 1989. ISBN 3-540-51701-4.
- [3] H. Kopetz. *Real-Time Systems: Design Principles for Distributed Embedded Applications*. Kluwer Academic Publishers, 1997. ISBN 0-7923-9894-7.
- [4] S. Mullender. *Distributed Systems*. Addison-Wesley, 1995. ISBN 0-201-62427-3.
- [5] J. Nehmer. *Experiences with Distributed Systems*, volume 309 of *Lecture Notes in Computer Science*. Springer-Verlag, 1987. ISBN 3-540-19333-2.
- [6] A. S. Tanenbaum and M. van Steen. *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2002. ISBN 0-13-088893-1.