

[Stand: 26.02.2009]

## Vorlesung 184.237 (VO 2.0) im WS 2008/2009

# Verteilte Systeme (Distributed Systems)



Beachten Sie auch die Informationen im [TUWIS](#), insbesondere Prüfungstermine, Prüfungsanmeldung und Anrechenbarkeit für diverse Studienrichtungen.



Beachten Sie auch die Informationen zur zugehörigen Laborübung [Distributed Systems Lab \(Verteilte Systeme LU\)](#), insbesondere zur verpflichtenden Anmeldung und Vorbesprechung.

[\[Ablauf\]](#) [\[Unterlagen\]](#) [\[Prüfung\]](#) [\[Vorbereitung und Fragenkatalog\]](#) [\[Kontakt\]](#)

## Ablauf

Die Vorlesung findet geblockt im Oktober und November statt. Die folgende Tabelle enthält die genauen Vorlesungszeiten und Inhalte:

Datum	Zeit	Ort	Inhalt
Di, 07.Oct	14:00-16:00	<a href="#">InfHS</a>	Überblick, <a href="#">Grundlagen und Konzepte</a>
Mi, 08.Oct	11:00-13:00	<a href="#">EI7</a>	<a href="#">Communication (1)</a>
Di, 14.Oct	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Communication (2)</a>
Mi, 15.Oct	11:00-13:00	<a href="#">EI7</a>	<a href="#">Operating System Support</a>
Di, 21.Oct	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Naming and Discovery</a>
Do, 23.Oct	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Clocks and Agreement</a>
Di, 28.Oct	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Consistency and Replication</a>
Mi, 29.Oct	11:00-13:00	<a href="#">EI7</a>	<a href="#">Consistency and Replication</a> (cont'd); <a href="#">Dependability and Fault Tolerance</a>
Di, 04.Nov	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Dependability and Fault Tolerance</a> (cont'd); Group Communication
Mi, 05.Nov	11:00-13:00	<a href="#">EI7</a>	<a href="#">Security</a>
Di, 11.Nov	14:00-16:00	<a href="#">InfHS</a>	<a href="#">Technolgy Overview</a> und Zusammenfassung

## Unterlagen

Die Kapitel- und Seitenangaben beziehen sich auf die (zweite) englische Ausgabe des Buches [Distributed Systems](#) von A. Tanenbaum und M. van Steen (2006), erhältlich in technischen Buchhandlungen oder z.B. bei [Amazon.com](#). Beachten Sie auch die [korrigierten Seiten](#), da einige

substanzielle Fehler des Buches dort korrigiert werden - insbesondere auf Seite 332 und 333. Die in Europa verfügbare englische Internationale Edition der zweiten Auflage (2008) ist laut Verlagsauskunft inhaltsgleich und kann z.B. bei [Amazon.de](http://www.amazon.de) erworben werden.

Die Vorlesungsfolien sind [online verfügbar](#). **Die Folien sind ohne Buch nicht zum Selbststudium oder als Prüfungsvorbereitung geeignet.**

Wenn Sie an einem Thema vertiefend interessiert sind, sei auf die [Originalliteratur](#) verwiesen. Weitere gute Bücher über Verteilte Systeme sind:

- Coulouris, J. Dollimore, T. Kindberg. [Distributed Systems: Concepts and Design](#), 4th ed., ISBN 0321-263-545, Addison-Wesley, 2005
- K. P. Birman, Reliable Distributed Systems, Springer, 2005, ISBN: 0-387-21509-3
- G. Alonso et al, Web Services, Springer, 2003, ISBN: 978-3540440086

## Prüfung

Die Prüfung ist grundsätzlich schriftlich abzulegen, es sind keine Unterlagen erlaubt. Die Prüfungsfragen werden auf Deutsch gestellt und können auf Deutsch oder Englisch beantwortet werden. Prüfungsrelevant ist der gesamte Vorlesungsstoff, der u.a. Fragenkatalog dient der Unterstützung der Vorbereitung auf Basis des Buches und der Vorlesungsfolien.

Die verpflichtende Anmeldung erfolgt über [TUWIS](#), wo Sie auch die genauen Termine finden. Sollten Sie kurz vor der Pr&uuml;fung beschließen doch nicht anzutreten, melden Sie sich bitte ab, damit Ihre Kollegen auf der Warteliste den freigewordenen Platz nutzen können. Die Bekanntgabe der Noten, Punkte pro Beispiel sowie des Termins zur Einsichtnahme erfolgt über TUWIS über die dort eingetragene Email-Adresse. **Achten Sie daher darauf, daß Ihre TUWIS Email-Adresse gültig ist.** Für die Zuordnung von Prozenten zur Note gilt die folgende Richtskala: Genügend (4) ab 50%, Befriedigend (3) ab 65%, Gut (2) ab 80%, Sehr Gut (1) ab 91%.

## Vorbereitung und Fragenkatalog

Es wird empfohlen, die u.a. Buchkapitel *vor* dem jeweiligen Vorlesungstermin als Vorbereitung zu lesen.

Der Fragenkatalog dient der Vorbereitung auf die schriftliche Vorlesungsprüfung. Sie müssen mit zehn Fragen rechnen, wobei mindestens sieben Fragen aus diesem Katalog stammen - allerdings können die Fragen mitunter aus den unten angeführten Fragestellungen etwas anders zusammengesetzt bzw. kombiniert sein. Bis zu drei Fragen können gänzlich neu sein. Die angegebenen Buchkapitel dienen der leichteren Zuordnung und stellen keine Einschränkung des Stoffumfanges dar - es gilt stets der gesamte in der Vorlesung vorgetragene Stoff.

## Grundlagen und Konzepte

Buch: Kap. 1 und 2 bis inkl. 2.2

Fragen:

1. Geben Sie eine charakterisierende Definition für ein "Verteiltes System" an. Nennen Sie die

wichtigsten Design-Ziele bzw. charakteristischen Eigenschaften von Verteilten Systemen. Stellen Sie weiters den Zusammenhang zu den typischen Fallstricken beim Entwurf verteilter Systeme her.

2. Was ist Transparenz? Beschreiben Sie die standardisierten Arten von Transparenz und erklären Sie den Zusammenhang zwischen den einzelnen Transparenz-Definitionen. Was ist der Nachteil von Transparenz?
3. Was versteht man unter "Openness"?
4. Erläutern Sie Probleme und Lösungsansätze für "Scalability".
5. Was versteht man unter der vertikalen Verteilung bzw. N-Schichten-Systemen? Diskutieren Sie dabei alle Grundvarianten von Client/Server-Systemen. Ist folglich ein Java Applet eher ein Thick Client oder ein Thin Client?
6. Was ist horizontale Verteilung? Mit welchen grundlegenden Design-Fragen müssen Sie sich beim Entwurf der horizontalen Verteilung eines Systems beschäftigen? Gibt es einen Zusammenhang zur vertikalen Verteilung?

## **Communication (1) - Middleware und RPC**

Buch: 2.1 (w.h.), 2.3, 2.4, 4.1, 4.2

Fragen:

1. Beschreiben Sie das ISO-OSI Modell der geschichteten Protokolle (Grundprinzip). Stellen Sie den Bezug zu den Internet-Protokollen (TCP/IP) her. Warum sind Transport-Layer Protokolle für Verteilte Systeme oft nicht ausreichend?
2. Was ist Middleware? Welche Anforderungen stellt man an Middleware? Welche Services soll Middleware bieten? Erläutern Sie den Zusammenhang zwischen Middleware und Architectural styles.
3. Wie kann man die Flexibilität der Middleware erhöhen sowie die Zusammenarbeit von Middleware und Anwendung effizienter gestalten? Erläutern Sie dabei die Grundprinzipien von Interceptoren, Adaptivität und Self-Management.
4. Erläutern Sie das Grundprinzip des Remote Procedure Call. Gehen Sie auf die Begriffe "client stub" und "server stub" näher ein.
5. Wie können Variablen bei Prozedur-Aufrufen grundsätzlich übergeben werden? Wie werden Sie bei RPC gehandhabt und welche Probleme gibt es dabei? Was versteht man in diesem Zusammenhang unter "parameter marshalling"?
6. Wie schreibt man für RPCs Client und Server und welche Rolle spielt dabei die IDL? Welche Ziele werden mit dem Einsatz einer IDL in einem verteilten System verfolgt? Was versteht man in diesem Zusammenhang unter "binding"?
7. Welche Arten von asynchronen RPCs gibt es? Geben Sie auch einen verallgemeinerten Überblick über verschiedene Typen der Kommunikation (persistent/transient bzw. synchron/asynchron).

## **Communication (2) - RMI und Messaging**

Buch: 4.3, 10.1.1, 10.3, 10.4

Fragen:

1. Erläutern Sie die Grundprinzipien verteilter Objekte sowie der Remote Object (bzw. Method) Invocation. Gehen Sie auf die Begriffe "proxy" und "skeleton" ein. Erklären Sie

- den Unterschied zwischen "Compile-time" und "Run-time" Objekten. Erklären Sie den Unterschied zwischen persistenten und transienten Objekten.
2. Wie funktioniert das Binding bei RMI? Welchen Zusammenhang gibt es zu den verschiedenen Arten, eine object reference zu implementieren. Vergleichen Sie (exemplarisch) CORBA und Java in Bezug auf Objektreferenzen.
  3. Was versteht man unter "static" und "dynamic" Invocation von verteilten Methoden? Geben Sie Beispiele an.
  4. Wie funktioniert die Parameterübergabe bei RMI? Gehen Sie dabei auf jene Eigenschaften der Objektorientierung ein, welche den Vorteil von RMI gegenüber RPC bewirken.
  5. Erläutern Sie die Grundprinzipien von Message-orientierter Kommunikation und gehen Sie auf CORBA Messaging exemplarisch ein. Beschreiben Sie zwei unterschiedliche Methoden, wie asynchrone Methodenaufrufe in CORBA Messaging erfolgen können.
  6. Was versteht man unter "Message-oriented Middleware MoM"? Erläutern Sie Modell und Architektur solcher "Message-Queueing"-Systeme. Erklären Sie die Primitivoperationen Put, Get, Poll und Notify eines Message-Queueing Systems. Diskutieren Sie Einsatzzwecke sowie Vor- und Nachteile - gehen Sie insbesondere auf den Begriff des Message Brokers und dessen Bedeutung für EAI ein.
  7. Erklären Sie "stream-oriented communication". Was ist "QoS" und inwiefern ist es für stream-oriented communication von Bedeutung?

## Operating System Support

Buch: 10.2.1, Kap. 3

Interessierte finden zum Thema Client in dem [Paper über die Entwicklung von User Interface Tools](#) weitere Informationen.

Interessierte finden in der [CORBA Spezifikation](#) im Kapitel 11 (POA - Portable Object Adapter) eine konkrete Umsetzung des allgemeinen Objekt-Adapter-Konzeptes.

Fragen:

1. Was ist der Unterschied zwischen Prozess und Thread? Was ist beim Einsatz von Multi-Threading zu beachten? Welche Bedeutung haben Threads in verteilten Systemen, insbesondere in Client/Server-Umgebungen?
2. Welche Aspekte Verteilter Systeme sind auf Client-Seite zu berücksichtigen? Wie werden User Interfaces in die Architektur Verteilter Systeme eingebunden? Wie können dabei verschiedene Arten der Transparenz unterstützt werden?
3. Geben Sie grundlegende Design-Entscheidungen für Server an und bewerten Sie diese. Gehen Sie auf den Unterschied zwischen stateful und stateless Servern genauer ein und geben Sie Beispiele an. Erläutern Sie anhand einer Skizze Architektur und Funktionsweise eines multi-threaded Servers (zB File- oder Web-Server). Was ist beim Einsatz von Multi-Threading vom Entwickler besonders zu beachten?
4. Was sind die Besonderheiten von Objekt-Servern? Welche Arten gibt es dabei für die "Invocation", also den Aufruf (evtl. auch die Aktivierung/Activation) eines Objektes auf Server-Seite (Policies hinsichtlich thread, code sharing, und object creation)? Was ist in diesem Zusammenhang ein Objekt-Adapter?
5. Erläutern Sie die wichtigsten Aspekte der Code Migration. Erklären Sie "strong mobility" und "weak mobility" und geben Sie für "weak mobility" ein Beispiel an.
6. Erläutern Sie das Konzept der Virtualisierung und in weiterer Folge deren Bedeutung für die Code Migration in heterogenen Umgebungen. Beschreiben Sie die zwei verschiedenen

Arten von Architekturen von "virtual machines".

## Naming and Discovery

Buch: Kap. 5

Fragen:

1. Erläutern Sie die Begriffe "Name", "Identifier", "Address" sowie den Bezug zwischen diesen Begriffen in der Praxis.
2. Was ist ein "Name Space"? Erläutern Sie das Grundprinzip des "Closure Mechanismus" anhand eines Beispiels (zB Unix File System).
3. Erklären Sie die Schichten der Verteilung von Name Spaces. Erläutern Sie die Einsatzmöglichkeiten von Replication und Caching in den verschiedenen Schichten. Erklären Sie verschiedene (hierarchische) Möglichkeiten der iterativen/rekursiven "name resolution".
4. Erläutern Sie das Domain Name System DNS, sowie den Ablauf bei der Namens-Auflösung anhand der DNS Database (Resource Records). Was ist reverse lookup? Was ist ein zone-transfer?
5. Was ist Directory Service bzw. "Attribute-based naming"? Beschreiben Sie den prinzipiellen Aufbau des X.500 Name Space sowie dessen LDAP Implementierung.
6. Wie funktioniert in flachen Namensräumen das Location Service? Geben Sie das Grundprinzip möglicher Lösungen an und gehen Sie dabei auch auf die Begriffe Mobility und Discovery ein. Erläutern Sie Vor- und Nachteile bei der Verwendung von "Forwarding Pointers". Wie funktionieren "Home-based approaches" für mobile Geräte?

## Clocks and Agreement

Buch: Kap. 6

[Buchauszug \(c\) Pearson/Prentice Hall - mit freundlicher Genehmigung des Verlages](#) (für die Frage zum "Global state")

Fragen:

1. Wozu braucht man Uhrensynchronisation? Erläutern Sie das NTP und den Berkeley Algorithmus. Was ist die Problematik bei der Synchronisation von Physical Clocks?
2. Was sind die Gründe für die Verwendung von Logical Clocks? Erklären Sie die Unterschiede zu den Physical Clocks. Was ist die "happened-before" Beziehung und wie funktionieren die "Lamport-Timestamps"?
3. Welchen Nachteil haben die Lamport-Timestamps und wie kann dieser durch Vector-Timestamps überwunden werden?
4. Wie funktioniert Distributed Mutual Exclusion. Wie verhalten sich verschiedene Algorithmen (centralized, distributed, token-ring) hinsichtlich Skalierbarkeit und Fehlertoleranz?
5. Sie sollen einen Dateiserver implementieren der mehrere Clients gleichzeitig bedienen kann. Nennen und beschreiben Sie einen (konkreten) Mechanismus der garantiert dass immer nur einer der Clients gleichzeitig eine Datei schreiben darf. Die anderen Clients sollen solange blockiert werden (egal ob schreibend oder lesend) bis der schreibende Client seine Arbeit beendet hat. Warum ist das sinnvoll?
6. Erläutern Sie den "Bully" und den "Ring"-Algorithmus für Election und vergleichen sie die

beiden hinsichtlich Fehlertoleranz. Warum sind diese Algorithmen für ad-hoc oder large-scale Systeme weniger geeignet und welche grundsätzlichen Lösungsansätze verfolgt man daher dort?

7. Welche Probleme gibt es bei der Ermittlung des "Global State" und wie können diese überwunden werden? Geben Sie zumindest einen Algorithmus an.

## Consistency and Replication

Buch: 7.1, 7.4, 7.5, 10.6, 4.5.2

Fragen:

1. Was sind die Hauptgründe für den Einsatz von Replikation in verteilten Systemen? In welcher Beziehung stehen Replikation und Skalierbarkeit zueinander? Erläutern Sie in diesem Zusammenhang verschiedene Varianten der Content Replication und des Content Placement.
2. Geben Sie verschiedene Möglichkeiten der Update Propagation (Content Distribution) an und bewerten Sie diese hinsichtlich Vor- und Nachteilen sowie Einsatzmöglichkeiten.
3. Erläutern Sie die Funktionsweise der "primary-based" Protokolle. Bewerten und vergleichen Sie die verschiedenen Arten.
4. Erläutern Sie die Funktionsweise der "replicated-write" Protokolle. Bewerten und vergleichen Sie die verschiedenen Arten. Welche Probleme können bei "Active Replication" auftreten? Erklären Sie "quorum-based" Replikationsprotokolle und geben Sie die Bedingungen an, um Read-Write und Write-Write Konflikte zu verhindern.
5. Welche Besonderheiten sind bei der Replikation von Objekten zu beachten? Erläutern Sie (inkl. genauer Skizze) wie man "Replication transparency" in Objektsystemen umsetzen könnte ("replicated invocation").
6. Was sind Epidemic Protocols. Welche Vor- und Nachteile haben diese? Erklären Sie "gossiping" ("rumor spreading") im Zusammenhang mit Replica update propagation. Erläutern Sie Vor- und Nachteile. Erklären Sie das Anti-Entropy Modell im Zusammenhang mit Replica update propagation. Erläutern Sie Vor- und Nachteile.

## Dependability and Fault Tolerance

Buch: 8.1 - 8.4

[Buchauszug \(c\) Pearson/Prentice Hall - mit freundlicher Genehmigung des Verlages](#) (für die Frage zum "Two army problem")

Fragen:

1. Erläutern Sie die grundlegenden Begriffe der Dependability: Nennen Sie die fünf wesentlichen Attribute (bzw. Requirements) eines "dependable system". Was ist der Unterschied zwischen Availability und Reliability? Erläutern Sie die "dependability threats" Failure, Error und Fault sowie den Zusammenhang zwischen den drei. Erläutern Sie "permanent", "transient" und "intermittent" faults anhand von Beispielen.
2. Geben Sie verschiedene Fehlermodelle ("failure models") an und diskutieren Sie diese. Wozu benötigt man überhaupt Fehlermodelle? Inwiefern ist es u.U. heikel zu spezifizieren, daß ein System "k-fault-tolerant" sein soll?
3. Wieso benötigt man Redundanz zur Maskierung von Fehlern? Welche Arten von Redundanz gibt es?

4. Erläutern Sie die Aussage des "two-army" Problems.
5. Erläutern Sie die Aussage der "Byzantinischen Generäle".
6. Erläutern Sie die Fehlerklassen in RPC-Client/server-Umgebungen. Gehen Sie besonders auf das "lost reply" Problem ein.
7. Was versteht man unter reliable bzw. ordered multicast (group communication) in statischen Gruppen von Prozessen? Was muss man bedenken, wenn sich die Gruppen dynamisch verändern können? Erläutern Sie das Prinzip des "atomic multicast" ("virtual synchrony").

## Security

Buch: Kap. 9, 12.8

Fragen:

1. Erläutern Sie die Definition von Security mit den Attributen Availability, Confidentiality und Integrity anhand von Beispielen. Beschreiben Sie jeweils vier Security Threats und Security Mechanisms.
2. Was ist der Secure Socket Layer (SSL, auch TLS genannt)? Positionieren Sie SSL/TLS im Internet Protokoll Stack.
3. Geben Sie eine Definition für "Cryptography" an und erläutern Sie die Funktionsweise von symmetrischen und asymmetrischen Verschlüsselungsverfahren. Gehen Sie dabei auch auf die spezifischen Vor- und Nachteile ein und geben Sie konkrete Beispiele für Algorithmen an.
4. Welche Eigenschaften erwartet man sich von einem "Secure Channel"? Geben Sie jeweils auch Beispiele für unerwünschte Effekte an. Erklären Sie wie sich zwei Kommunikationsteilnehmer basierend auf Public Key Cryptography gegenseitig authentifizieren können.
5. Erklären Sie, wie sich zwei Kommunikationsteilnehmer basierend auf symmetrischen Schlüsseln gegenseitig authentifizieren können. Zeigen Sie auch auf, welche Probleme entstehen können, wenn ein Protokoll falsch "optimiert" wird.
6. Was ist eine digitale Signatur, welche Garantien bietet sie und wie funktioniert sie? Was ist eine Hash Funktion und welche Eigenschaften muss sie erfüllen, damit sie im Rahmen einer digitalen Signatur eingesetzt werden kann.
7. Was ist ein Key Distribution Centre (KDC), wozu wird es eingesetzt und welchen Vorteil bietet es? Geben Sie ein konkretes Verfahren zur gegenseitigen Authentifizierung von Kommunikationsteilnehmern mit Hilfe eines KDC an und beschreiben Sie für jeden Schritt, wer sich wem gegenüber bereits authentifiziert hat.

## Technology Overview

Buch: 10.1.2, Überblick Kap. 12 und 13

Fragen:

1. Ordnen Sie in der folgenden Taxonomie den vier Typen von Koordinationsmodellen jeweils zwei konkrete Techniken, Beispiele oder Systeme zu.

Temporal	
Coupled	Uncoupled

Referential	Coupled	(a) Direct	(b) Mailbox
	Uncoupled	(c) Meeting oriented	(d) Generative Communication

2. Erläutern Sie das Grundprinzip von publish/subscribe als Kommunikations-/Koordinationsmechanismus. Worin bestehen die Möglichkeiten aber auch die Probleme dieses Mechanismus (v.a. dessen Implementierung). Erläutern Sie weiters JINI und JavaSpaces als konkrete Technologien.
3. Was ist ein Web-Server? Wozu dient das Hypertext Transfer Protokoll HTTP? Nennen Sie zwei HTTP Operationen. Diskutieren Sie die Funktionsweise eines CGI-Programms.

## Kontakt

**Kontakt und E-Mail:** [Dr. Karl. M. Göschka](#)

**Sprechstunde:** Unmittelbar nach der Vorlesung oder nach Vereinbarung per E-Mail.

This space is left blank intentionally.