

LEHRSTUHL FÜR  
ALLG. BWL UND WIRTSCHAFTSINFORMATIK  
UNIV.-PROF. DR. HERBERT KARGL

*Schwickert, Axel C.; Grimbs, Michael G.*

**Grundlagen des  
Client/Server-Konzepts**

ARBEITSPAPIERE WI  
Nr. 1/1996

---

Schriftleitung:  
Dr. rer. pol. Axel C. Schwickert

# Information

---

- Reihe:** Arbeitspapiere WI
- Herausgeber:** Univ.-Prof. Dr. Axel C. Schwickert  
Professur für BWL und Wirtschaftsinformatik  
Justus-Liebig-Universität Gießen  
Fachbereich Wirtschaftswissenschaften  
Licher Straße 70  
D – 35394 Gießen  
Telefon (0 64 1) 99-22611  
Telefax (0 64 1) 99-22619  
eMail: [Axel.Schwickert@wirtschaft.uni-giessen.de](mailto:Axel.Schwickert@wirtschaft.uni-giessen.de)  
<http://wi.uni-giessen.de>
- Bis Ende des Jahres 2000 lag die Herausgeberschaft bei:
- Lehrstuhl für Allg. BWL und Wirtschaftsinformatik  
Johannes Gutenberg-Universität Mainz  
Fachbereich Rechts- und Wirtschaftswissenschaften  
Welderweg 9  
D - 55099 Mainz
- Ziele:** Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.
- Zielgruppen:** Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IuK-Management und Praktiker in Unternehmen.
- Quellen:** Die Arbeitspapiere entstanden aus Forschungsarbeiten, Diplom-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr- und Vortragsveranstaltungen des Lehrstuhls für Allg. Betriebswirtschaftslehre und Wirtschaftsinformatik Univ. Prof. Dr. Herbert Kargl an der Johannes Gutenberg-Universität Mainz.
- Hinweise:** Wir nehmen Ihre Anregungen und Kritik zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.  
Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit dem Herausgeber (Gießen) unter obiger Adresse Kontakt auf.  
Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe und deren Bezug erhalten Sie auf dem Schlußblatt eines jeden Arbeitspapiers und auf der Web Site des Lehrstuhls unter der Adresse <http://wi.uni-giessen.de>

# Arbeitspapiere WI Nr. 1/1996

---

**Autoren:** Schwickert, Axel C.; Grimbs, Michael G.

**Titel:** Grundlagen des Client/Server-Konzepts

**Zitation:** Schwickert, Axel C.; Grimbs, Michael G.: Grundlagen des Client/Server-Konzepts, in: Arbeitspapiere WI, Nr. 1/1996, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 1996.

**Kurzfassung:** Das Client/Server-Konzept dient der Unterstützung eines betriebswirtschaftlich bedingten, organisatorischen Wandels, weg von vertikalen, funktionsorientierten Strukturen und hin zu horizontaler Prozeßorientierung. Da eine Organisation die Gesamtheit aller Regeln und damit ein logisches Konstrukt repräsentiert, muß daraus folgen, daß ein organisationsangepaßtes Client/Server-Konzept nicht allein durch Hardware-Systeme beschrieben sein kann. Unter Client/Server wird in neueren Veröffentlichungen von der überwiegenden Mehrheit der Autoren ein logisches Software-Konzept verstanden, das einer gewandelten Unternehmensorganisation angepaßt sein und sie unterstützen soll. Trotz dieses gemeinsamen Grundverständnisses existiert noch keine einheitliche und umfassende Auffassung über das Client/Server-Konzept. In der Praxis hat sich als ein gemeinsamer Nenner die Sichtweise der Gartner Group mit der geschilderten fünfstufigen Client/Server-Prozeßstruktur durchgesetzt, die sich an der Geschäftsprozeßlogik orientiert.

**Schlüsselwörter:** Client/Server, Prozesse, Schichten, horizontale/vertikale Verteilung, Daten, Applikation, Präsentation, Inter-Prozeß-Kommunikation, Remote Procedure Call, Stored Procedures, Datenbanksystem, Trigger, Duplizierung, Partitionierung

## Inhaltsverzeichnis

1	Rahmendefinition des Client/Server-Konzepts .....	3
2	Client/Server als logisches Konzept .....	4
2.1	Prozesse und Schichten von IuK-Systemen.....	4
2.2	Verteilte Präsentation.....	7
2.3	Entfernte Präsentation.....	9
2.4	Kooperative Verarbeitung.....	10
2.5	Entfernte Datenbank .....	13
2.6	Verteilte Datenbank .....	15
3	Zusammenfassung: Client/Server-Konzept .....	17
	Literaturverzeichnis.....	18

## 1 Rahmendefinition des Client/Server-Konzepts

Die Ansicht, daß es sich beim dem Begriff Client/Server um ein **logisches Konzept** handelt, wird in der Literatur nicht von allen Autoren geteilt. Es existiert eine große Bandbreite bezüglich der Definitionen von Client/Server, die zunächst skizziert werden soll, bevor das Konzept definitorisch konsensfähig dargestellt wird.

Für Trimmer<sup>1</sup> sind die Charakteristika eines Client/Server-Konzeptes die gekoppelten PC- oder UNIX-Systeme, die über ein LAN oder ähnliche Verkabelungsformen verbunden sind. Damit beschränkt sich Trimmer auf die **reine Hardware-Sicht** von Client/Server und bildet das eine Extrem der Definitionsskala. Für Inmon<sup>2</sup> ist, wie bei Trimmer, die Hardware-Vernetzung das Hauptkriterium für Client/Server. Er nimmt die Abgrenzung des Client/Server-Konzeptes von der monolithischen Großrechnerarchitektur dahingehend vor, daß er den Fluß der Rohdaten zwischen den einzelnen Knoten und deren lokale Verarbeitung als entscheidendes Merkmal im Client/Server-Konzept einstuft.

Sinha<sup>3</sup> geht von einer Hardware-Orientierung ab und definiert Client/Server als ein **Verarbeitungsparadigma**. Bauer<sup>4</sup> folgt dieser Betrachtungsweise und richtet sein Augenmerk auf die Art der Anwendung, wenn er feststellt, daß die Verbindung von PCs über ein LAN noch nichts über deren Einsatzform aussage. So kann innerhalb eines Netzwerkes auf jedem Knoten eine monolithische, das heißt, eine in sich völlig geschlossene und von anderen Programmen unabhängige, Anwendung laufen. In diesem Sinn argumentiert auch Meyer<sup>5</sup>, wenn er anführt, daß die klassische Software, welche aus den drei Teilen Transaktionsmonitor, Applikation und Datenbank besteht, monolithisch ist. Dies trifft nicht nur auf die typischen COBOL-Großrechnerlösungen zu, sondern auch auf moderne PC-Lösungen. Beide Autoren argumentieren daher, daß das Client/Server-Konzept erst wirklich neue Möglichkeiten bietet, wenn man es als Software-Konzept versteht. Diese neuen Möglichkeiten manifestieren sich in einer flexiblen **Nachbildung der verteilten, betriebswirtschaftlichen Strukturen** durch Informations- und Kommunikations-Systeme (IuK-Systeme).

---

1 Vgl. Trimmer, D.: Downsizing - Strategies for Success in the Modern Computer World, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 62.

2 Vgl. Inmon, W. H.: Client/Server-Anwendungen - Planung und Entwicklung, Übersetzung aus dem Englischen: Dobrowolski, P., Springer-Verlag, Berlin u. a. 1993, S. 7.

3 Vgl. Sinha, A.: Client-Server Computing, in: Communications of the ACM, Bd. 35, Nr. 7, 1992, S. 78.

4 Vgl. Bauer, M.: Mehr als ein Haufen vernetzter PCs - Chancen und Probleme von Client/Server-Architekturen, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 19.

5 Vgl. Meyer, H.-M.: Softwarearchitekturen für verteilte Verarbeitung, in: Hansen, W.-R. (Hrsg.), Client-Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 73 f.

Die Meinung, daß Client/Server ein logisches Konzept darstellt, hat sich heute bei der Mehrzahl der Autoren prinzipiell durchgesetzt. Als Gegenpol zur Hardware-Sicht wird dabei oftmals von einer "Software-Architektur"<sup>6</sup> gesprochen. Als Grundlage der folgenden Ausführungen wird im Sinne einer Rahmendefinition **das Client/Server-Konzept als ein logisches (Software-) Konzept verstanden, welches Hardware-technisch umgesetzt sein will**. Zur Begriffsabgrenzung wird ein auf Basis dieses Client/Server-Konzeptes auch technisch realisiertes IuK-System hier mit dem Terminus Client/Server-System versehen.

## 2 Client/Server als logisches Konzept

### 2.1 Prozesse und Schichten von IuK-Systemen

Ein Unternehmen läßt sich aus betriebswirtschaftlicher Sicht in die Teile Komponenten **Mitarbeiter, betriebliche Funktionen und Daten** gliedern (siehe Abb. 1). Die betrieblichen Funktionen im Sinne Kosiols stellen die betriebswirtschaftlichen Aufgaben des Unternehmens dar; die Daten werden von diesen Funktionen be- und verarbeitet; die Mitarbeiter wenden die Funktionen auf die Daten an. Um dieser betriebswirtschaftlichen Sichtweise zu folgen, erscheint es zweckmäßig, auch IuK-Systeme spiegelbildlich in drei Schichten einzuteilen:<sup>7</sup>

- ❑ **Datenschicht:** Datenhaltung und Datenmanipulation
- ❑ **Applikationsschicht:** Funktions- und Ablauflogik
- ❑ **Präsentationsschicht:** Schnittstelle zum Anwender

Die **Definition von Geschäftsprozessen** legt fest, welche Mitarbeiter, Funktionen und Daten an bestimmten Vorgängen beteiligt sind. Demzufolge werden IuK-Systeme, die einzelne Geschäftsprozesse, Vorgangsketten oder Vorgänge unterstützen, aus entsprechenden Präsentations-, Applikations- und Datenkomponenten zusammengestellt (siehe Abb. 1).

Ein solches IuK-System kann aus logischer und technischer Sicht als **monolithischer Block** realisiert werden: Auf einem zentralen Rechner läuft in diesem Fall ein Programm ab, das sowohl den ihm zugeordneten Datenbestand verwaltet, als auch alle Verarbeitungsschritte beinhaltet und zusätzlich die Output-Steuerungen vornimmt. Veränderungen der betriebswirtschaftlichen Vorgangsketten haben dann jedoch zwingend zur Folge, daß grundlegende Veränderungen des gesamten, hochintegrierten IuK-Systems vorzunehmen sind.

---

6 Vgl. Bauer, M.: Client/Server: Mehr eine Software-Frage, in: Online, 12/1993, S. 45.

7 Vgl. Krcmar, H.: Informationsmanagement und Informationssystem-Architekturen - Vorteile und Risiken von Client-Server-Architekturen aus der Sicht des Informationsmanagements, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client-Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 15.

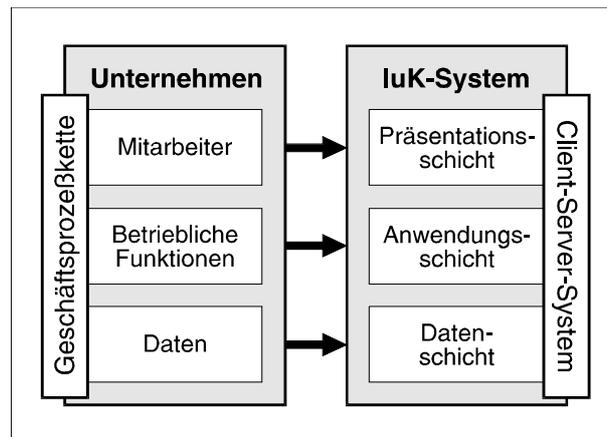


Abb. 1: Geschäftsprozessketten werden durch Client/Server-Systeme abgebildet

**Das Client/Server-Konzept postuliert die verteilte Implementierung** von Präsentations-, Applikations- und Datenkomponente nach Maßgabe der zugrundeliegenden betriebswirtschaftlichen Aufgaben auf eine Client- und eine Server-Seite. Ein erstes, leicht nachvollziehbares Beispiel zeigt Abb. 2.

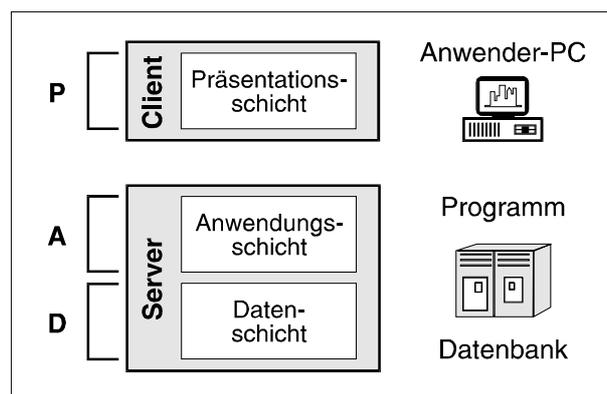


Abb. 2: Logische Verteilung der drei Schichten auf Client- und Server-Prozess

Auf der Client-Seite findet hier nur der Kontakt mit dem Anwender des IuK-Systems statt. Zum Beispiel nutzt ein Anwender zur Ausführung einer fachlichen Aufgabe die Maus- und Fensterfunktionalität einer grafischen Benutzeroberfläche, die ihm der PC auf seinem Schreibtisch zur Verfügung stellt. Die Funktionen und Daten zur Erledigung der fachlichen Aufgabe, also das Programm, das die Bedienungsanweisungen entgegennimmt und auf die betroffenen Daten ausführt, werden von einem zweiten Rechner auf der Server-Seite vorgehalten. Auf der Client-Seite wird durch einen logischen Prozess, dem **Client-Prozess**, ein Dienst nachgefragt (und später empfangen), auf der Server-Seite wird durch einen zweiten logischen Prozess, dem **Server-Prozess**, ein Dienst erbracht. Im Beispiel aus Abb. 2 verläuft die Client/Server-Grenze zwischen der Präsentationsschicht auf der einen und der Applikations- und Datenschicht auf der anderen Seite. **Die Positionierung dieser Grenze wird bei der Planung eines Client/Server-Systems**

**primär logisch vorgenommen.** Das heißt, daß die jeweils zu erfüllende betriebswirtschaftliche Aufgabe zunächst ausschlaggebend dafür ist, welche Schichtkomponenten welcher Seite (welchem der beiden logischen Prozesse) zuzuordnen sind. Im obigen Beispiel wäre es wenig sinnvoll, die Applikationsschicht, sprich das Programm, zusätzlich zur Präsentationskomponente auf der Client-Seite zu positionieren, wenn seine Verarbeitungsfunktion auch für andere Clients (zum Beispiel andere Arbeitsplatzrechner) zentral und permanent verfügbar sein soll. **Die logische Verteilung der drei Schichtkomponenten auf Client- und Server-Prozeß muß nicht zwingend auch technisch nachvollzogen werden.** Das bedeutet, daß die beiden Prozesse zwar - wie beschrieben - auf zwei Rechnern als getrennte Software-Produkte implementiert sein können, wobei die Bezeichnung der beiden Rechner der logischen Rollenverteilung in Client- und Server-Prozeß folgt. In Multitasking-/ Multiusing-fähigen Systemen können aber sowohl Client- als auch Server-Prozeß in verschiedenen Adreßräumen eines einzigen Rechners ablaufen. Diesem Rechner werden dann gleichzeitig die beiden Rollen Client und Server zuteil. Die Hardware-orientierte Interpretation des Server-Begriffs aus der Welt der Netzwerktechnik (LAN, WAN) weicht im Client/Server-Konzept der primär dienstleistungsorientierten Interpretation. In welcher Hardware-Landschaft sich die logischen Clients und Server niederschlagen, ist zumindest bei der Planung eines Client/Server-Systems von nachrangigem Interesse.

Ein in der Praxis weit verbreitetes und akzeptiertes Verfahren der Rollenzuweisung geht auf die Gartner Group zurück (siehe Abb. 3). Die drei Schichten Präsentation, Applikation und Daten werden eindeutig der Client- beziehungsweise Server-Seite zugeordnet. Von **horizontaler Verteilung** spricht man, wenn die Trennung innerhalb einer Schicht vollzogen wird, das heißt, ein Teil der Schicht dem Client-, der komplementierende Teil der gleichen Schicht dem Server-Prozeß zugeordnet wird. Verläuft die Trennlinie zwischen zwei Schichten, wird dies als **vertikale Verteilung bezeichnet**.<sup>8</sup> Im allgemeinen unterscheidet man fünf Kombinationsformen: **Verteilte Präsentation, Entfernte Präsentation, Kooperative Verarbeitung, Entfernte Datenbank und Verteilte Datenbank.** Vereinzelt findet die Form der Verteilten Präsentation keine Berücksichtigung, oder es werden, wie bei Krcmar, weitere Formen gebildet.<sup>9</sup> Der Prozeß, welcher dem Anwender in Form der Präsentationsschnittstelle am nächsten liegt, wird im Schema der Gartner Group als Client bezeichnet. Dies mag auf der Tatsache beruhen, daß aufgrund der Hardware-geprägten Definition des Client/Server-Begriffs in den 80er Jahren die Desktops als Clients und die Hintergrundsysteme als Server bezeichnet wurden.<sup>10</sup> Die in Abb. 3 neben dem Client/Server-Konzept dargestellte konventionelle Lösung wurde der Vollständigkeit halber und zur Kontrastierung mit dem Client/Server-Konzept aufgenommen. Das Gefüge aus Präsentations-, Anwendungs- und Datenschicht ist hier lo-

---

8 Vgl. Petzold, H. J.; Schmitt, H.-J.: Verteilte Anwendungen auf der Basis von Client-Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 170, 1993, S. 81.

9 Vgl. Krcmar, H., a. a. O., S. 15 unterscheidet noch die Form der "Totalen Verteilung", welche aber als Sonderform der Verteilten Datenbank gesehen werden kann.

10 Vgl. Buck-Emden, R.: Chancen für kommerzielle Lösungen in neuer Qualität - Kürzere Antwortzeiten und erweiterte Dialogmöglichkeiten steigern den Nutzen, in: Supplement zur ComputerWorld Expo, Computerwoche, Jg. 20, Nr. 43 vom 22.10.1993, S. 8.

gisch nicht getrennt. Die Software ist monolithisch und übernimmt sowohl die Datenhaltung, als auch die Bearbeitung und Präsentation.

Da die **Rollenverteilung der Gartner Group** in der Literatur nicht nur den höchsten Verbreitungs- und Akzeptanzgrad hat, sondern auch eine strukturierte Beschreibung des Client/Server-Konzeptes auf der Basis dieser Sichtweise eindeutig durchführbar ist, soll sie **im folgenden als prinzipielle Beschreibungsgrundlage** dienen. Ausgeführt werden vorrangig die fünf "Reinformen" der Verteilung. Bei der Realisierung des Client/Server-Konzeptes in der Praxis werden aber sicherlich Kombinationen dieser Reinformen Anwendung finden, um die individuell verschiedenen Geschäftsprozesse bestmöglich abzubilden. Wenn anschließend von Clients und Servern gesprochen wird, so sind damit durch Software realisierte, logische Prozesse gemeint, die nicht physisch verteilt sein müssen, dies aber durchaus sein können.

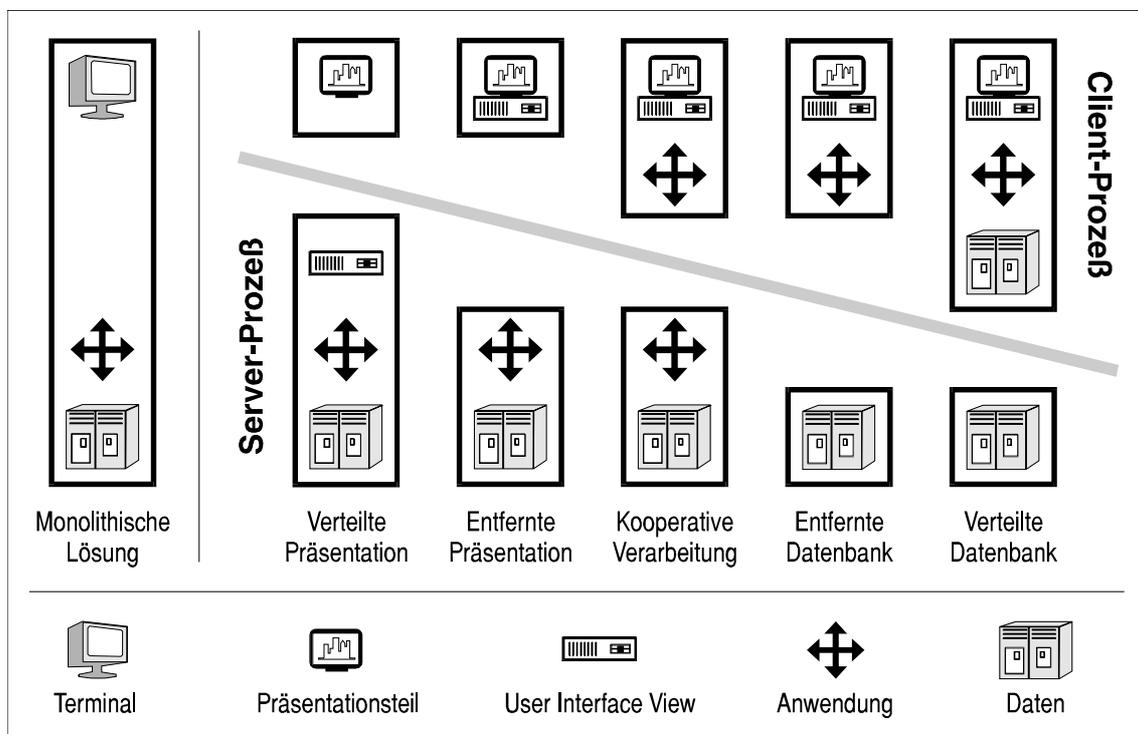


Abb. 3: Das Client/Server-Konzept<sup>11</sup>

## 2.2 Verteilte Präsentation

Bei der **Verteilten Präsentation** erfolgt innerhalb der Präsentationsschicht eine **horizontale Verteilung in den User Interface View und den Präsentationsteil**. Der User Interface View auf der Server-Seite hat die Aufgabe, die abstrakte Anwendungsfunktio-

<sup>11</sup> Vgl. Gartner Group, in: Jost, U.: Beim Downsizing können OLTP-Produkte hilfreich sein - Datenbanksysteme machen die OLTP-Monitore überflüssig, in: Computerwoche, Jg. 20, Heft Nr. 24 vom 11.06.1993, S. 43, verändert.

nalität der Applikationsschicht für den Präsentationsteil auf der Client-Seite zur Darstellung vorzubereiten.<sup>12</sup> Dazu hält der User Interface View Funktionen wie das Erzeugen von Fenstern oder Zeichen von grafischen Primitiven bereit und baut die Verbindung zum Präsentationsteil auf.<sup>13</sup> Der Präsentationsteil wiederum hat die Aufgabe, eine Darstellung auf dem Bildschirm zu ermöglichen. Dazu interpretiert er die Nachrichten des User Interface View und stellt seine Oberflächenressourcen wie zum Beispiel Fenster, Fonts und Cursor bereit.<sup>14</sup> Eine solche Teilung der Präsentationsschicht ist beispielsweise in X-Windows realisiert und wird in Abb. 4 dargestellt.

Der Nomenklatur von Abb. 3 folgend wird der Präsentationsteil dem Client-Prozeß zugeordnet, der User Interface View zusammen mit der Anwendungs- und Datenschicht dem Server-Prozeß. Löst man sich vom Konzept der Gartner Group mit definierten Seiten für Client- und Server-Prozeß, kann der Präsentationsteil als Präsentations-Server aufgefaßt werden, der dem User Interface View und dem Rest der Anwendung seine Darstellungsdienste liefert. Diese Namensgebung resultiert dann aus der Betrachtung der logischen Folge von Dienstleistungsbeziehungen zwischen den Prozessen und wird im konkreten Fall von X-Windows auch so durchgeführt. D. h., bei X-Windows wird der Präsentationsteil als X-Server bezeichnet, der User Interface View als X-Client.

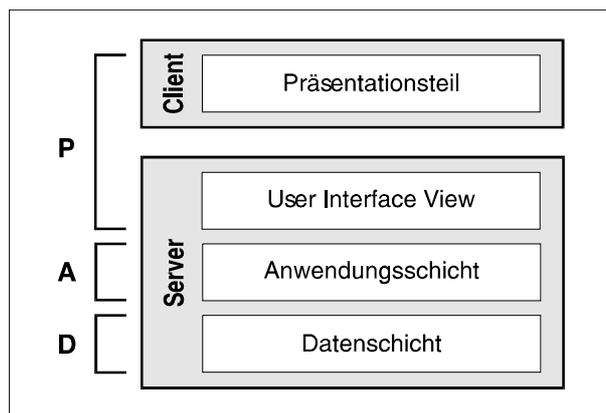


Abb. 4: Verteilte Präsentation

Da die Trennlinie zwischen Client und Server bei der Verteilten Präsentation innerhalb einer Schicht, hier der Präsentationsschicht, verläuft, wird von horizontaler Verteilung gesprochen. Dabei ist diese **Trennlinie logischer Natur**, das heißt, die Software-Produkte, die jeweils die Client- und Server-Funktionalität erfüllen, können auf mehreren intelligenten Workstations **physisch verteilt** vorliegen, müssen dies aber nicht.

Ein **Vorteil der Verteilten Präsentation** liegt darin, daß der vollkommen abgetrennt laufende Präsentationsteil die Voraussetzung schafft, unterschiedliche Applikationen, die auf völlig heterogenen Server-Plattformen ablaufen können, dem Anwender in gleicher Form auf dem Bildschirm zu präsentieren. Dazu wird jeweils ein User Interface

<sup>12</sup> Vgl. Meyer, H.-M., a. a. O., S. 74 f.

<sup>13</sup> Vgl. Meyer, H.-M., a. a. O., S. 92.

<sup>14</sup> Vgl. Meyer, H.-M., a. a. O., S. 91.

View in die Server-Prozesse integriert, die den Präsentationsteilen eine einheitliche Darstellung erlauben. Die Aufbereitung muß aber nicht zwangsläufig von einer grafischen Oberfläche im Client entgegengenommen werden. Der User Interface View kann mehrere Sichten auf eine Applikation ermöglichen, zum Beispiel eine grafische oder eine alphanumerische. Es kann fallweise entschieden werden, welche Form der Darstellung problemangemessen ist.

Mit der Verteilten Präsentation ist es möglich, **Anwendungen auf der Oberflächenebene so zu integrieren**, daß der Anwender den Eindruck gewinnt, es handle sich um eine einzige Anwendung. Beispielsweise könnte im Rahmen einer solchen Integration das interne Marketinginformationssystem eines Unternehmens mit dem Datenbankrecherchesystem eines externen Dienstleisters an der Oberfläche integriert werden, ohne daß eine Abstimmung bezüglich Anwendungsprozessen und Datenstrukturen notwendig ist. Da die Verteilte Präsentation auf diese Weise die Integration der Anwendungs- oder Datenprozesse außen vorläßt, kann mit Meyer von einer **”scheinbaren Integration”**<sup>15</sup> gesprochen werden, die sich ausschließlich in der Oberflächenschicht zeigt.

### 2.3 Entfernte Präsentation

Bei der **Entfernten Präsentation**, wie sie in Abb. 5 schematisch gezeigt wird, verläuft die logische **Trennlinie zwischen der Präsentationsschicht**, die dem Client-Prozeß zugerechnet wird **und der Anwendungs- und Datenschicht**, die auf der Server-Seite angesiedelt sind. Da im Gegensatz zur Verteilten Präsentation die gesamte Präsentationsschicht auf den Client-Prozeß ausgelagert wird, spricht man hier von **vertikaler Verteilung**.

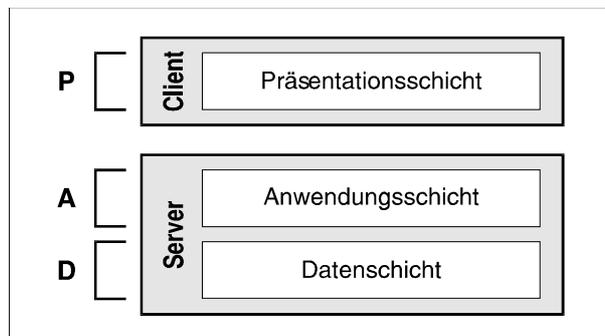


Abb. 5: Entfernte Präsentation

Die vertikale Verteilung - ist ebenso wie die horizontale - logischer Natur, was bedeutet, daß Client- und Server-Prozeß nicht zwangsläufig auf zwei verschiedenen Rechnern realisiert sein müssen. Beispiele für Präsentationsschichten sind Windows 3.x, Windows 95 und der OS/2 Presentation Manager.

<sup>15</sup> Meyer, H.-M., a. a. O., S. 83.

Der Ausgangspunkt der Namensgebung für diese Verteilungsart ist der Server, das heißt aus Sicht von Anwendungs- und Datenschicht ist die Präsentation entfernt im Client-Prozeß implementiert. Die Verteilung wird im allgemeinen so interpretiert, daß die Präsentationsschicht den Client-Prozeß darstellt, der sich verschiedener Server-Prozesse bedient, in denen die Applikations- und Datendienste liegen. Wie bei der Verteilten Präsentation bietet sich auch hier die Möglichkeit, die Präsentationsschicht als Präsentations-Server zu interpretieren, der der Anwendungs- und Datenschicht seine Grafikdienste anbietet.

Die **Vorteile dieser Verteilungsform** ergeben sich aus der Möglichkeit, **jedem einzelnen Anwender die Schnittstelle anzubieten, die sich für seine Arbeit als ergonomisch sinnvoll erweist**. Dies muß nicht unbedingt eine grafische Oberfläche (Graphical User Interface, GUI) sein.<sup>16</sup> Meyer<sup>17</sup> nennt in diesem Zusammenhang das Beispiel des Managers, der eine Anwendung nur sporadisch nutzt und daher eine rein grafische, intuitiv bedienbare Benutzerschnittstelle in Anspruch nimmt, sowie das Beispiel des Datenerfassers, für den die alphanumerische Darstellung die optimale Schnittstelle bietet.

**Nachteilig** wirkt sich zur Zeit noch das **Fehlen von weitreichenden Standards** zur Gestaltung der Entfernten Präsentation aus, da die Schnittstellen von verbreiteten Oberflächen wie Windows 3.x oder OS/2 Presentation Manager auf der Client-Seite nicht herstellerübergreifend genormt sind. Eine Realisierung dieser Verteilungsform ist daher mit Unwägbarkeiten für die Unternehmen bezüglich Zukunftssicherheit und Investitionsschutz verbunden. Kommt zum Beispiel unternehmensweit eine Präsentationsoberfläche zum Einsatz, deren Schnittstellenspezifikationen nicht einem sich entwickelnden Standard entsprechen, so muß mit erheblichem Anpassungs- und Wartungsaufwand für die bereits im Unternehmen in dieser Verteilungsvariante realisierten Anwendungen gerechnet werden.

## 2.4 Kooperative Verarbeitung

Bei der **Kooperativen Verarbeitung** (auch: Verteilte Anwendung, Verteilte Funktion) verläuft die **Trennlinie zwischen Client- und Server-Prozeß horizontal innerhalb der Anwendungsschicht**, in der die Geschäftslogik umgesetzt ist. Dabei sind im Client-Prozeß die Präsentations- und ein Teil der Anwendungsschicht, im Server-Prozeß der zweite Teil der Anwendungsschicht und die Datenschicht realisiert (siehe Abb. 6).

Die innerbetriebliche Funktionsaufteilung in zentrale und dezentrale Aufgaben schafft eine Strukturvorgabe, die durch das IuK-System nachvollzogen wird. Sollen zum Beispiel Immobilieninvestitionen beurteilt werden, wendet man unter anderem die Kapitalwertmethode an: Die Funktionalität der Kapitalwertmethode kann dezentral in dem Teil der Anwendungsschicht implementiert sein, der zum Client-Prozeß gehört. Wenn aber auch ein anderer Client-Prozeß für eine Beurteilung, von zum Beispiel Finanzanlagen,

---

<sup>16</sup> Vgl. Renaud, P. E.: Introduction To Client/Server Systems - A Practical Guide for Systems Professionals, John Wiley & Sons Inc., New York u. a. 1993, S. 8.

<sup>17</sup> Vgl. Meyer, H.-M., a. a. O., S. 84.

die Kapitalwertmethode benötigt, integriert man diesen Algorithmus in den zentralen Teil der Anwendungsschicht des Server-Prozesses.

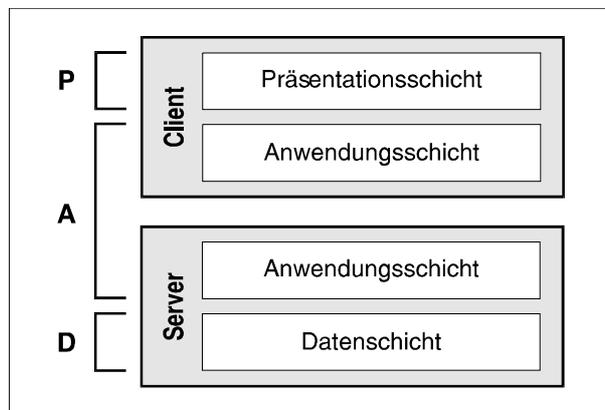


Abb. 6: Kooperative Verarbeitung

Client- und Server-Prozeß sind nicht ständig verbunden. Für den Fall, daß einer der Clients den Dienst des Servers in Anspruch nimmt, muß eine **Inter-Prozeß-Kommunikation (IPC)** aufgebaut werden. Dafür bieten sich mehrere Möglichkeiten an: Im einfachsten Fall liegen die Client- und Server-Prozesse im Adressraum eines einzigen Rechners. Der Informationsaustausch kann dann im gemeinsamen Speicherbereich (**Shared Memory**) vonstatten gehen, zum Beispiel über eine gemeinsam genutzte Variable. Da die Prozesse in der Praxis jedoch meist physisch verteilt sind, wird der Nachrichtenaustausch entweder in Form der "nachrichten-basierten Kommunikation" oder durch Prozeßfernaufrufe (Remote Procedure Call, RPC) realisiert.<sup>18</sup> Bei der **nachrichten-basierten Kommunikation** werden, per unidirektionaler Übertragung, Nachrichten zwischen den physisch verteilten Prozessen über ein Netz ausgetauscht. Diese Variante erfordert jedoch eine netzspezifische Programmierleistung zur Synchronisation der Prozesse.<sup>19</sup> Beim nachfolgend beschriebenen **Remote Procedure Call (RPC)** hingegen finden Kontrollfluß- und Datenübergabe synchron und netzunabhängig zwischen den Client- und Server-Prozessen statt. Seine Anwendung ähnelt der Benutzung einer lokalen Subroutine und machen den **RPC zu der zentralen Gestaltungskomponente der Kooperativen Verarbeitung**.

Wie in Abb. 7 dargestellt, beinhalten sowohl der Client-, als auch der Server-Teil bei der Nutzung von RPCs eine **Anschlußroutine (Stub)** und eine **Kommunikationskomponente**. Der Server-Prozeß bietet den Clients seine Dienste, im Beispielfall den Algorithmus zur Berechnung des Kapitalwertes, an. Aufgrund der Komplexität von Client/Server-Systemen wird er dies meist mit Hilfe eines sogenannten "**Service-Brokers**"

<sup>18</sup> Vgl. Heinrich, L. J.; Lehner, F.; Roithmayr, F.: Informations- und Kommunikationstechnik für Betriebswirte und Wirtschaftsinformatiker, 3., vollständig überarbeitete und erweiterte Auflage, Oldenbourg-Verlag, München, Wien 1993, S. 77.

<sup>19</sup> Vgl. Schill, A.: Basismechanismen und Architekturen für Client-Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 174, 1993, S. 14.

tun.<sup>20</sup> Dies ist ein eigenständiger Server-Prozeß, der die Dienste verschiedener Server an nachfragende Clients weitervermittelt.

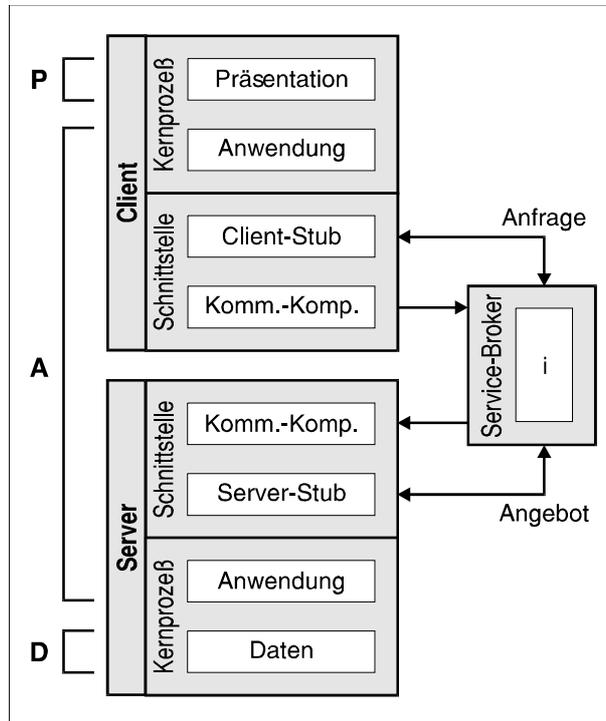


Abb. 7: Service-Broker und RPC

Dem Service-Broker ist bekannt, welche Dienste auf welchem Server vorgehalten werden. Der Client-Prozeß, der im Rahmen der Investitionsbeurteilung von Immobilien die Funktion der Kapitalwertberechnung benötigt, sie selbst aber nicht implementiert hat, ruft nun diesen Dienst zunächst bei der Client-Anschlußroutine (Client-Stub) auf. Die Client-Stub lokalisiert mit Hilfe des Service-Brokers den nachgefragten Server-Prozeß.<sup>21</sup> Dabei kann der Broker (Synonym: Agent) bei zu vielen, gleichzeitig eintreffenden Anfragen seitens der Clients, zum Engpaßfaktor werden. Die Client-Stub kodiert die Anforderung an die Server-Prozedur, das heißt, die Parameter der Investitionsrechnung werden in ein für die Übertragung geeignetes Format gebracht. Anschließend überträgt die Client-Kommunikationskomponente den Aufruf (Request-Message) zur Server-Kommunikationskomponente. Bis zum Empfang der Antwort vom Server ist der Client blockiert und im Wartezustand. Der Server-Prozeß wird aktiviert, berechnet den Kapitalwert und sendet das Ergebnis in einer Reply-Message via Server-Stub und Kommunikationskomponente wieder an den Client zurück. Daraufhin kann dieser im Programmablauf fortfahren.

20 Vgl. Schill, A.: Basismechanismen ..., a. a. O., S. 13.

21 Vgl. Schill, A.: Remote Procedure Call: Fortgeschrittene Konzepte und Systeme - ein Überblick. Teil 1: Grundlagen, in: Informatik-Spektrum, Band 15, 1992, S. 80.

Die Stärken der Kooperativen Verarbeitung sind eng mit den **Stärken des RPCs** verknüpft. Im Vergleich zur nachrichten-basierten Kommunikation wird nicht bei jedem RPC eine (eventuell sogar programmiererabhängige), netzspezifische Programmierleistung benötigt. Die Inanspruchnahme des Netzwerkes erfolgt über die **koordiniert vorliegenden Kommunikationskomponenten**. Da die Implementierung eines RPCs keine Detailkenntnisse der Applikationsentwickler über Kommunikationsmechanismen voraussetzt, gilt er im Vergleich zur nachrichtenbasierten Kommunikation als **prinzipiell robust und sicher**.

Ein weiterer Vorteil des RPCs liegt in der Möglichkeit, bestimmte, von verschiedenen Client-Prozessen benötigte Funktionen, wie den im Beispiel angeführten Algorithmus zur Kapitalwertberechnung, zu **kapseln** und auszulagern. Solche **zentralen Server-Funktionen**, die idealerweise eine Multi-Client-Fähigkeit<sup>22</sup> besitzen sollten, können dann auf einer bezüglich der Performanz **optimierten Hardware-Umgebung** zum Ablauf gebracht und durch den RPC wie ein Unterprogramm aufgerufen werden.

Da Client- und Server-Prozeß nur in konkreten Auftragssituationen in Beziehung stehen, danach aber wieder voneinander unabhängig sind, stellt **jeder RPC eine eigene Transaktion** dar. Das bedeutet, daß für jede Kommunikation zwischen den Prozessen erneut eine Verbindung aufgebaut werden muß. Somit entsteht im Vergleich zu monolithischen Programmen ein **erhöhter Kommunikationsaufwand**. Renaud<sup>23</sup> verweist ferner darauf, daß die RPCs zwar viele Nutzungsvorzüge aufweisen, ihre **Implementierung jedoch anspruchsvoll** ist. Im Planungsvorfeld muß eine ganze Reihe von Fragen, wie beispielsweise die Behandlung multipler Client-Prozesse bei einem Server oder die Autorisierung der Clients für einen Server-Prozeß, geklärt werden.

## 2.5 Entfernte Datenbank

Bei der **Entfernten Datenbank** (auch: Entfernte oder Ausgelagerte Datenhaltung) wird, wie aus Abb. 8 ersichtlich, die **Datenschicht komplett der Server-Seite zugeordnet; die Präsentations- und Anwendungsschicht liegen auf der Client-Seite**. Diese **vertikale Verteilungsform** entspricht dem klassischen Konzept des Datenbank-Servers, wie er auch Hardware-technisch oft realisiert ist.

Bezüglich der Namensgebung war bei der Entfernten Präsentation aus Sicht des Server-Prozesses der Client-Prozeß (in Form der Präsentationsschicht) entfernt. Ebenso wird bei der Entfernten Datenbank der Server-Prozeß aus Sicht der im Client-Prozeß zusammengefaßten Präsentations- und Anwendungsschicht als entfernt eingestuft.

Das Faktum, daß die Datenschicht vollständig auf der Server-Seite liegt, sagt aber noch nichts über die physische und logische Verteilung der Datenbestände auf der Server-Seite oder die Auftragsbeziehungen zwischen Client und Server aus. So können die **Da-**

---

22 Vgl. Renaud, P. E., a. a. O., S. 177: Unter Multi-Client-Fähigkeit wird verstanden, daß ein Server-Prozeß für mehrere Client-Prozesse zur Verfügung steht und nicht während der Bearbeitung eines RPCs blockiert ist.

23 Vgl. Renaud, P. E., a. a. O., S. 177.

**tenbestände** in mehreren Servern verteilt vorliegen, sie **können dupliziert oder partitioniert sein**, oder es können ein oder mehrere Server den Clients als Dienstleister gegenüberreten. Diese Möglichkeiten sollen im Rahmen der Verteilten Datenbank (siehe Abschnitt 2.6) näher beschrieben werden. An dieser Stelle soll nur die prinzipielle Auftragsbeziehung zwischen Client- und Server-Prozeß wiedergegeben, sowie die Konzepte der Stored Procedures und Trigger näher erläutert werden.

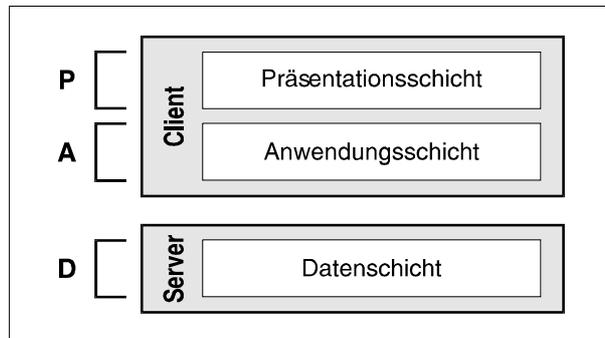


Abb. 8: Entfernte Datenbank

Der Server-Prozeß stellt an der definierten Schnittstelle zum Client diesem sein Datenbankmanagementsystem (DBMS) zur Verfügung. Darauf greift der Client mit Hilfe der Datendefinitionssprache (DDL) und der Datenmanipulationssprache (DML), bei relationalen Datenbanken über SQL, zu.<sup>24</sup>

Jenz unterscheidet bei dieser Form der **primitiv verteilten Datenbanken** zwischen "Remote Request" und "Remote Unit of Work".<sup>25</sup> Da in der Praxis die relationalen Datenbanken für die Verwaltung von Massendaten noch keine durchgängige Verbreitung gefunden haben, werden zum Beispiel Daten aus zentralen hierarchischen oder Netzwerk-Datenbanken in lokale, relationale Datenbanken übertragen. Beim **Remote Request** wird durch SQL-Abfragen nur lesend auf die relationalen Datenbanken zugegriffen. Die Praxisrelevanz dieser Form liegt daher auch nicht in operativen Anwendungssystemen, sondern vorrangig in selektierenden und auswertenden Entscheidungsunterstützungssystemen.

Bei der **Remote Unit of Work** wird lesend und ändernd auf die Daten zugegriffen. Dies hat zwei Folgen: Zum einen muß im Fehlerfall gewährleistet sein, daß die gesamte Datenbank-Transaktion reversibel gestaltet ist. Zum andern tritt bei der Verwendung von SQL unter dem Gesichtspunkt der Minimierung des Nachrichtenverkehrs zwischen den Prozessen eine nicht kommunikationsoptimale Situation ein: Jeder SQL-Befehl findet in

24 Vgl. Ambichl, E.; Heinrich, L. J.: Ergebnisse einer Leistungsbewertung von Client-Server-Architekturen, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client-Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 61.

25 Jenz, D. E.: Datenbanken in Netzwerken - Die Komplexität beherrschen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Nr. 174, 1993, S. 45.

mindestens zwei Kommunikationsschritten statt, dem Request und dem Reply.<sup>26</sup> Man faßt daher häufig gleichartig benutzte SQL-Sequenzen zu logischen Einheiten zusammen, die ganze Request/Reply-Transaktionen darstellen. Diese Prozeduren zur Datenmanipulation werden als **Stored Procedures** bezeichnet und vorcompiliert in der Datenbank selbst abgelegt. Die mit einem eindeutigen Namen versehenen Stored Procedures können vom Client über das DBMS abgerufen werden.<sup>27</sup>

Um Aufgaben, wie das Gewährleisten referenzieller Integrität, Duplikatenüberwachung oder Fehleroperationen wahrzunehmen, bedient man sich sogenannter **Trigger**. Dies sind spezielle Stored Procedures, die nicht direkt aufgerufen werden können, sondern mit den Tabellen der Datenbank verbunden sind und automatisch bei Operationen auf die Tabellen ausgeführt werden. Da in den Stored Procedures ein Teil der Verarbeitungslogik enthalten ist, entsteht eine Zwitterform zwischen Entfernter Datenbank und Kooperativer Verarbeitung.

Jenz<sup>28</sup> verweist in diesem Zusammenhang auf die Problematik von **Standardisierungslücken in SQL**. Viele Datenbankhersteller füllen diese Lücken mit proprietären Konzepten und Verfahren. In diesem Zusammenhang sei beispielhaft für eine ganze Reihe von Problemfeldern auf uneinheitliche Datentypen und Fehlercodes verwiesen. Hansen<sup>29</sup> ergänzt, daß nicht nur das Netz der Standards in SQL lückenhaft ist, sondern auch die verabschiedeten Standards selbst nicht exakt und eindeutig sind. Ein Sicherheitsproblem sehen Briem und Denzel<sup>30</sup> darin, daß es seitens der Clients möglich ist, individuell auf das DBMS im Server-Prozeß zuzugreifen und Datenmanipulationen durchzuführen. Im Gegensatz dazu läßt sich bei der Kooperativen Verarbeitung das DBMS durch die im Server-Prozeß integrierten Teile der Anwendungsschicht zentral gegen Zugriffe kapseln.

## 2.6 Verteilte Datenbank

Die logische Verteilung von Client- und Server-Prozeß verläuft bei der **Verteilten Datenbank horizontal innerhalb der Datenschicht**. Hier sind nicht nur die Präsentations- und die Anwendungsschicht dem Client-Prozeß zugeordnet, sondern auch ein Teil der Datenschicht. Dabei wurde hier bewußt der Begriff der Verteilten Datenbank und nicht etwa derjenige der "verteilten Datenhaltung" gewählt. Die "verteilte Datenhaltung" schränkt den Blick zu sehr auf die Splittung eines physikalischen Datenbestandes

---

26 Vgl. Briem, J.; Denzel, B.: Online Transaction Processing mit Client-Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Nr. 174, 1993, S. 33.

27 Vgl. Niemann, K. D.: Abschied von monolithischen Software-Strukturen - Entwicklung verteilter Anwendungen für Client/Server-Architekturen, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 21.

28 Vgl. Jenz, D. E., a. a. O., S. 45 ff.

29 Vgl. Hansen, W.-R.: Client-Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, in: Hansen, W.-R. (Hrsg.), Client-Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 31.

30 Vgl. Briem, J.; Denzel, B., a. a. O., S. 33.

ein. Eine Datenbank ist jedoch nicht als monolithischer Block anzusehen, sondern läßt sich in Datenbankmanagementsystem (DBMS) und Datenbestand differenzieren. Das DBMS besteht wiederum aus Datendefinitionssprache (DDL), Datenmanipulationssprache (DML) und SQL-Schnittstelle; eine Menge von Tabellen beschreibt bei relationalen Datenbanken den Datenbestand. Aufgrund dieser Zergliederung existiert ein weites Spektrum von Möglichkeiten, wie die horizontale Aufteilung der Datenschicht auf Client- und Server-Prozeß vorgenommen werden kann. Da im Rahmen dieser Arbeit die Verteilungsproblematik von Datenbanken nicht ausführlich beschrieben werden kann, sollen im folgenden nur die zwei Hauptvarianten und die damit grundsätzlich verbundenen Probleme besprochen werden:

- **Logischer Funktionalitätsschnitt** innerhalb des Database Management Systems (DBMS)
- **Physischer Schnitt** innerhalb des Datenbestandes

Die erste Variante (siehe Abb. 9) ist analog der bisher diskutierten horizontalen Verteilungsansätze Verteilte Präsentation und Kooperative Verarbeitung konzipiert. Dort wurde die Funktionalität der respektiven Schichten zwischen Client- und Server-Prozeß aufgeteilt. So wird auch hier die **Funktionalität des DBMS auf Client und Server verteilt**. Dies kann aufgrund der Tatsache geschehen, daß es zum Beispiel nicht nötig oder nicht sinnvoll ist, auf jedem Client die DDL und DML vorzuhalten.

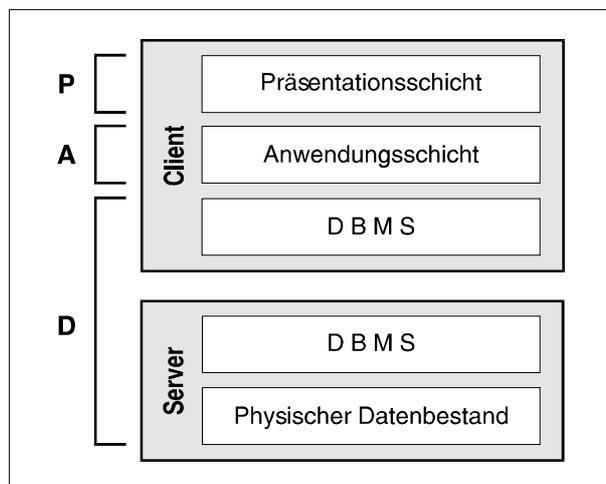


Abb. 9: Verteilte Datenbank mit verteiltem DBMS

Bei der zweiten Verteilungsmöglichkeit eines physischen Bestandsschnittes liegt **ein Datenbestand sowohl auf der Client- als auch der Server-Seite** vor. An dieser Stelle sollen nur die beiden Reinformen duplizierte und partitionierte Daten Erwähnung finden. Unter einer Duplikation von Daten wird die vollständige Vervielfachung eines Datenbestandes verstanden, unter Partitionierung das überschneidungsfreie Aufteilen der Daten. Die Partitionierung kann im Sinne einer Selektion horizontal durchgeführt werden, oder vertikal im Sinne einer Projektion.

Ein Beispiel, in dem sich die **Duplizierung von Daten** anbietet, ist die Verteilung der Devisenwechselkursstabelle von Banken. In dieser zentral gepflegten Tabelle werden die für alle Bankfilialen gültigen Tagesverrechnungskurse gehalten. Die Duplikate der Devisenwechselkursstabelle müssen in allen Filialen vollständig und schnell verfügbar sein. Die lokalen Anwendungen greifen dabei nur lesend auf sie zu, Änderungen werden nicht vorgenommen. Ein tägliches Update der Duplikate ist ausreichend und kann auch außerhalb des Geschäftsbetriebes durchgeführt werden.

Ein Beispiel für die **überschneidungsfreie Partitionierung** ist eine Personaldatenbank, die horizontal so partitioniert wird, daß Datensätze, die primär für eine bestimmte Filiale interessant sind, einzig auf dem Client jener Filiale vorgehalten werden. Jeder Datenbankknoten erhält somit nur die für ihn relevanten Datensätze. Dabei sollte die Aufteilung der organisatorischen Logik folgen. Das bedeutet, daß Daten, die in erster Linie auf der Client-Seite Verwendung finden, auch genau dieser zuzuteilen sind.

### 3 Zusammenfassung: Client/Server-Konzept

Unter Client/Server wird in neueren Veröffentlichungen von der überwiegenden Mehrheit der Autoren ein logisches Software-Konzept verstanden, das einer gewandelten Unternehmensorganisation angepaßt sein und sie unterstützen soll. Trotz dieses gemeinsamen Grundverständnisses existiert noch keine einheitliche und umfassende Auffassung über das Client/Server-Konzept. In der Praxis hat sich als ein gemeinsamer Nenner die Sichtweise der Gartner Group mit der geschilderten fünfstufigen Client/Server-Prozeßstruktur durchgesetzt, die sich an der Geschäftsprozeßlogik orientiert.

## Literaturverzeichnis

- Ambichl, E.; Heinrich, L. J.: Ergebnisse einer Leistungsbewertung von Client/Server-Architekturen, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client/Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 55-101.
- Bauer, M.: Client/Server: Mehr eine Software-Frage, in: Online, 12/1993, S. 45-47.
- Bauer, M.: Mehr als ein Haufen vernetzter PCs - Chancen und Probleme von Client/Server-Architekturen, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 19-20.
- Briem, J.; Denzel, B.: Online Transaction Processing mit Client/Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Nr. 174, 1993, S. 25-41.
- Buck-Emden, R.: Chancen für kommerzielle Lösungen in neuer Qualität - Kürzere Antwortzeiten und erweiterte Dialogmöglichkeiten steigern den Nutzen, in: Supplement zur ComputerWorld Expo, Computerwoche, Jg. 20, Nr. 43 vom 22.10.1993, S. 8-9.
- Gartner Group, in: Jost, U.: Beim Downsizing können OLTP-Produkte hilfreich sein - Datenbanksysteme machen die OLTP-Monitore überflüssig, in: Computerwoche, Jg. 20, Heft Nr. 24 vom 11.06.1993, S. 43
- Hansen, W.-R.: Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, in: Hansen, W.-R. (Hrsg.), Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 13.67.
- Heinrich, L. J.; Lehner, F.; Roithmayr, F.: Informations- und Kommunikationstechnik für Betriebswirte und Wirtschaftsinformatiker, 3., vollständig überarbeitete und erweiterte Auflage, Oldenbourg-Verlag, München, Wien 1993.
- Inmon, W. H.: Client/Server-Anwendungen - Planung und Entwicklung, Übersetzung aus dem Englischen: Dobrowolski, P., Springer-Verlag, Berlin u. a. 1993.
- Jenz, D. E.: Datenbanken in Netzwerken - Die Komplexität beherrschen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Nr. 174, 1993, S. 42-57.
- Krcmar, H.: Informationsmanagement und Informationssystem-Architekturen - Vorteile und Risiken von Client/Server-Architekturen aus der Sicht des Informationsmanagements, in: Krcmar, H.; Strasburger, H., (Hrsg.), Client/Server-Architekturen - Herausforderung an das Informationsmanagement, 1. Auflage, AIT Angewandte Informationstechnik, Hallbergmoos 1993, S. 9-29.
- Meyer, H.-M.: Softwarearchitekturen für verteilte Verarbeitung, in: Hansen, W.-R. (Hrsg.), Client/Server-Architektur - Grundlagen und Herstellerkonzepte für Downsizing und Rightsizing, Addison-Wesley Publishing Company, Bonn u. a. 1993, S. 69-116.
- Niemann, K. D.: Abschied von monolithischen Software-Strukturen - Entwicklung verteilter Anwendungen für Client/Server-Architekturen, in: Computerwoche Extra, Nr. 4 vom 17.09.1993, S. 21-25.
- Petzold, H. J.; Schmitt, H.-J.: Verteilte Anwendungen auf der Basis von Client/Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 170, 1993, S. 79-92.
- Renaud, P. E.: Introduction To Client/Server Systems - A Practical Guide for Systems Professionals, John Wiley & Sons Inc., New York u. a. 1993.
- Schill, A.: Basismechanismen und Architekturen für Client/Server-Architekturen, in: HMD-Theorie und Praxis der Wirtschaftsinformatik, Jg. 30, Heft Nr. 174, 1993, S. 8-24.
- Schill, A.: Remote Procedure Call: Fortgeschrittene Konzepte und Systeme - ein Überblick. Teil 1: Grundlagen, in: Informatik-Spektrum, Band 15, 1992, S. 79-87.
- Sinha, A.: Client/Server Computing, in: Communications of the ACM, Bd. 35, Nr. 7, 1992, S. 77-98.

# Bisher erschienen

Stand: Dezember 2000 – Den aktuellen Stand der Reihe erfahren  
Sie über unsere Web Site unter <http://wi.uni-giessen.de>

---

Nr. 1/1996	Grundlagen des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 2/1996	Wettbewerbs- und Organisationsrelevanz des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 3/1996	Realisierungsaspekte des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 4/1996	Der Geschäftsprozeß als formaler Prozeß - Definition, Eigenschaften, Arten.....	Schwicker/Fischer
Nr. 5/1996	Manuelle und elektronische Vorgangsteuerung.....	Schwicker/Rey
Nr. 6/1996	Das Internet im Unternehmen - Neue Chancen und Risiken.....	Schwicker/Ramp
Nr. 7/1996	HTML und Java im World Wide Web.....	Gröning/Schwicker
Nr. 8/1996	Electronic-Payment-Systeme im Internet.....	Schwicker/Franke
Nr. 9/1996	Von der Prozeßorientierung zum Workflow-Management - Teil 1: Grundgedanken, Kernelemente, Kritik.....	Maurer
Nr. 10/1996	Von der Prozeßorientierung zum Workflow-Management - Teil 2: Prozeßmanagement und Workflow.....	Maurer
Nr. 11/1996	Informationelle Unhygiene im Internet.....	Schwicker/Dietrich/Klein
Nr. 12/1996	Towards the theory of Virtual Organisations: A description of their formation and figure.....	Appel/Behr
Nr. 1/1997	Der Wandel von der DV-Abteilung zum IT-Profitcenter: Mehr als eine Umorganisation.....	Kargl
Nr. 2/1997	Der Online-Markt - Abgrenzung, Bestandteile, Kenngrößen.....	Schwicker/Pörtner
Nr. 3/1997	Netzwerkmanagement, OSI Framework und Internet SNMP.....	Klein/Schwicker
Nr. 4/1997	Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht.....	Strecker/Schwicker
Nr. 5/1997	Sachzielintegration bei Prozeßgestaltungsmaßnahmen.....	Delnef
Nr. 6/1997	HTML, Java, ActiveX - Strukturen und Zusammenhänge.....	Schwicker/Dandl
Nr. 7/1997	Lotus Notes als Plattform für die Informationsversorgung von Beratungsunternehmen.....	Appel/Schwaab
Nr. 8/1997	Web Site Engineering - Modelltheoretische und methodische Erfahrungen aus der Praxis.....	Schwicker
Nr. 9/1997	Kritische Anmerkungen zur Prozeßorientierung.....	Maurer/Schwicker
Nr. 10/1997	Künstliche Neuronale Netze - Aufbau und Funktionsweise.....	Strecker
Nr. 11/1997	Workflow-Management-Systeme in virtuellen Unternehmen.....	Maurer/Schramke
Nr. 12/1997	CORBA-basierte Workflow-Architekturen - Die objektorientierte Kernanwendung der Bausparkasse Mainz AG.....	Maurer
Nr. 1/1998	Ökonomische Analyse Elektronischer Märkte.....	Steyer
Nr. 2/1998	Demokratiopolitische Potentiale des Internet in Deutschland.....	Muzic/Schwicker
Nr. 3/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 1).....	Delnef
Nr. 4/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 2).....	Delnef
Nr. 5/1998	Betriebswirtschaftlich-organisatorische Aspekte der Telearbeit.....	Polak
Nr. 6/1998	Das Controlling des Outsourcings von IV-Leistungen.....	Jäger-Goy
Nr. 7/1998	Eine kritische Beurteilung des Outsourcings von IV-Leistungen.....	Jäger-Goy
Nr. 8/1998	Online-Monitoring - Gewinnung und Verwertung von Online-Daten.....	Guba/Gebert
Nr. 9/1998	GUI - Graphical User Interface.....	Maul
Nr. 10/1998	Institutionenökonomische Grundlagen und Implikationen für Electronic Business.....	Schwicker
Nr. 11/1998	Zur Charakterisierung des Konstrukts "Web Site".....	Schwicker
Nr. 12/1998	Web Site Engineering - Ein Komponentenmodell.....	Schwicker
Nr. 1/1999	Requirements Engineering im Web Site Engineering – Einordnung und Grundlagen.....	Schwicker/Wild
Nr. 2/1999	Electronic Commerce auf lokalen Märkten.....	Schwicker/Lüders
Nr. 3/1999	Intranet-basiertes Workgroup Computing.....	Kunow/Schwicker
Nr. 4/1999	Web-Portale: Stand und Entwicklungstendenzen.....	Schumacher/Schwicker
Nr. 5/1999	Web Site Security.....	Schwicker/Häusler
Nr. 6/1999	Wissensmanagement - Grundlagen und IT-Instrumentarium.....	Gaßen
Nr. 7/1999	Web Site Controlling.....	Schwicker/Beiser
Nr. 8/1999	Web Site Promotion.....	Schwicker/Arnold
Nr. 9/1999	Dokumenten-Management-Systeme – Eine Einführung.....	Dandl
Nr. 10/1999	Sicherheit von eBusiness-Anwendungen – Eine Fallstudie.....	Harper/Schwicker
Nr. 11/1999	Innovative Führungsinstrumente für die Informationsverarbeitung.....	Jäger-Goy
Nr. 12/1999	Objektorientierte Prozeßmodellierung mit der UML und EPK.....	Dandl
Nr. 1/2000	Total Cost of Ownership (TCO) – Ein Überblick.....	Wild/Herges
Nr. 2/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen.....	Franke/Sulzbach
Nr. 3/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 2: Der Einsatz im Unternehmen.....	Franke/Sulzbach
Nr. 4/2000	Web-Site-spezifisches Requirements Engineering – Ein Formalisierungsansatz.....	Wild/Schwicker
Nr. 5/2000	Elektronische Marktplätze – Formen, Beteiligte, Zutrittsbarrieren.....	Schwicker/Pfeiffer
Nr. 6/2000	Web Site Monitoring – Teil 1: Einordnung, Handlungsebenen, Adressaten.....	Schwicker/Wendt
Nr. 7/2000	Web Site Monitoring – Teil 2: Datenquellen, Web-Logfile-Analyse, Logfile-Analyzer.....	Schwicker/Wendt
Nr. 8/2000	Controlling-Kennzahlen für Web Sites.....	Schwicker/Wendt
Nr. 9/2000	eUniversity – Web-Site-Generierung und Content Management für Hochschuleinrichtungen.....	Schwicker/Ostheimer/Franke

---

# Bestellung (bitte kopieren, ausfüllen, zusenden/zufaxen)

**Adressat:** Professur für BWL und Wirtschaftsinformatik  
 Fachbereich Wirtschaftswissenschaften  
 Licher Straße 70  
 D – 35394 Gießen  
 Telefax: (0 641 ) 99-22619

**Hiermit bestelle ich gegen Rechnung die angegebenen Arbeitspapiere zu einem Kostenbeitrag von DM 10,- pro Exemplar (MwSt. entfällt) zzgl. DM 5,- Versandkosten pro Sendung.**

Nr.	An
1/1996	
2/1996	
3/1996	
4/1996	
5/1996	
6/1996	
7/1996	
8/1996	
9/1996	
10/1996	
11/1996	
12/1996	

Nr.	An
1/1997	
2/1997	
3/1997	
4/1997	
5/1997	
6/1997	
7/1997	
8/1997	
9/1997	
10/1997	
11/1997	
12/1997	

Nr.	Anz
1/1998	
2/1998	
3/1998	
4/1998	
5/1998	
6/1998	
7/1998	
8/1998	
9/1998	
10/1998	
11/1998	
12/1998	

Nr.	Anz
1/1999	
2/1999	
3/1999	
4/1999	
5/1999	
6/1999	
7/1999	
8/1999	
9/1999	
10/1999	
11/1999	
12/1999	

Nr.	Anz
1/2000	
2/2000	
3/2000	
4/2000	
5/2000	
6/2000	
7/2000	
8/2000	
9/2000	

**Absender:**

Organisation

Abteilung

Nachname, Vorname

Straße

Plz/Ort

Telefon

Telefax

eMail

Ort, Datum

Unterschrift