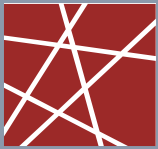




# Motivierende Fragen

- ❑ Welche Dienste soll eine Middleware bereitstellen?
- ❑ Wie funktioniert ein entfernter Prozeduraufruf?
- ❑ Was ist beim Aufruf entfernter Methoden zu beachten?
- ❑ Welche Fehler können beim entfernten Prozeduraufruf auftreten, und wie kann mit diesen Fehlern umgegangen werden?
- ❑ Was ist beim Aufruf entfernter Methoden zu beachten?



# Grundlagen: Rechnernetze und Verteilte Systeme

## Kapitel 9: Verteilte Systeme

Prof. Dr.-Ing. Georg Carle  
Lehrstuhl für Netzarchitekturen und Netzdienste  
Technische Universität München  
[carle@net.in.tum.de](mailto:carle@net.in.tum.de)  
<http://www.net.in.tum.de>

Acknowledgements: Prof. Dr. Wolfgang Kuchlin, Tübingen





1. Einführung und Motivation
  - Bedeutung, Beispiele
2. Begriffswelt und Standards
  - Dienst, Protokoll, Standardisierung
3. Direktverbindungsnetze
  - Fehlererkennung, Protokolle
  - Ethernet
4. Vermittlung
  - Vermittlungsprinzipien
  - Wegwahlverfahren
5. Internet-Protokolle
  - IP, ARP, DHCP, ICMP
  - Routing-Protokolle
6. Transportprotokolle
  - UDP, TCP
7. Verkehrssteuerung
  - Kriterien, Mechanismen
  - Verkehrssteuerung im Internet
8. Anwendungsorientierte Protokolle und Mechanismen
  - Netzmanagement
  - DNS, SMTP, HTTP
9. **Verteilte Systeme**
  - **Middleware**
  - **RPC, RMI**
  - **Web Services**
10. Netzsicherheit
  - Kryptographische Mechanismen und Dienste
  - Protokolle mit sicheren Diensten: IPSec etc.
  - Firewalls, Intrusion Detection
11. Nachrichtentechnik
  - Daten, Signal, Medien, Physik
12. Bitübertragungsschicht
  - Codierung
  - Modems

## Kapitel 9 - Teil 1

### 9.1 Grundlagen

### 9.2 Middleware

### 9.3 RPC

### 9.4 RMI

## Kapitel 9 - Teil 2

### 9.5 Service Oriented Architectures

### 9.6 Corba

### 9.7 Web-Anwendungen

### 9.8 HTML und XML

### 9.9 Web Services



## 9.1 Definition eines Verteilten Systems

- Ein verteiltes System ist eine Menge voneinander unabhängiger Computer, die dem Benutzer wie ein einzelnes, kohärentes System erscheinen.

[Tanenbaum, van Steen: Verteilte Systeme, Pearson Studium, 2003]



## Eigenschaften Verteilter Systeme

- ❑ Unterschiede zwischen den verschiedenen Computern werden verborgen.
- ❑ Benutzer und Applikationen können auf konsistente und einheitliche Weise mit dem verteilten System zusammenarbeiten.
- ❑ Verteilte Systeme sollen erweiterbar und skalierbar sein.

## Beispiele

- ❑ Netzwerk aus Workstations mit gemeinsamen Dateidiensten und gemeinsamer Benutzerverwaltung
- ❑ Informationssystem für Arbeitsabläufe
- ❑ World Wide Web



# Ziele für Verteilte Systeme

- ❑ Benutzer und Ressourcen verbinden
  - Den Benutzern ermöglichen, auf entfernte Ressourcen zuzugreifen
  - Unterstützung für kontrollierte gemeinsame Benutzung
- ❑ Transparenz
  - Zugriff – verbirgt Unterschiede in der Datendarstellung
  - Position – verbirgt Ort der Ressource
  - Migration – verbirgt Möglichkeit, Ressource an anderen Ort zu verschieben
  - Relokation – verbirgt Verschiebung von Ressource während Nutzung
  - Replikation – verbirgt, dass eine Ressource repliziert ist
  - Nebenläufigkeit – verbirgt gleichzeitige Nutzung konkurrierender Benutzer
  - Fehler – verbirgt Ausfall und Wiederherstellung einer Ressource
  - Persistenz – verbirgt Speicherart (Hauptspeicher oder Festplatte)
- ❑ Offenheit
  - Vollständige Schnittstellenspezifikation (⇒ Schnittstellendefinitionssprache IDL – Interface Description Language)
- ❑ Skalierbarkeit



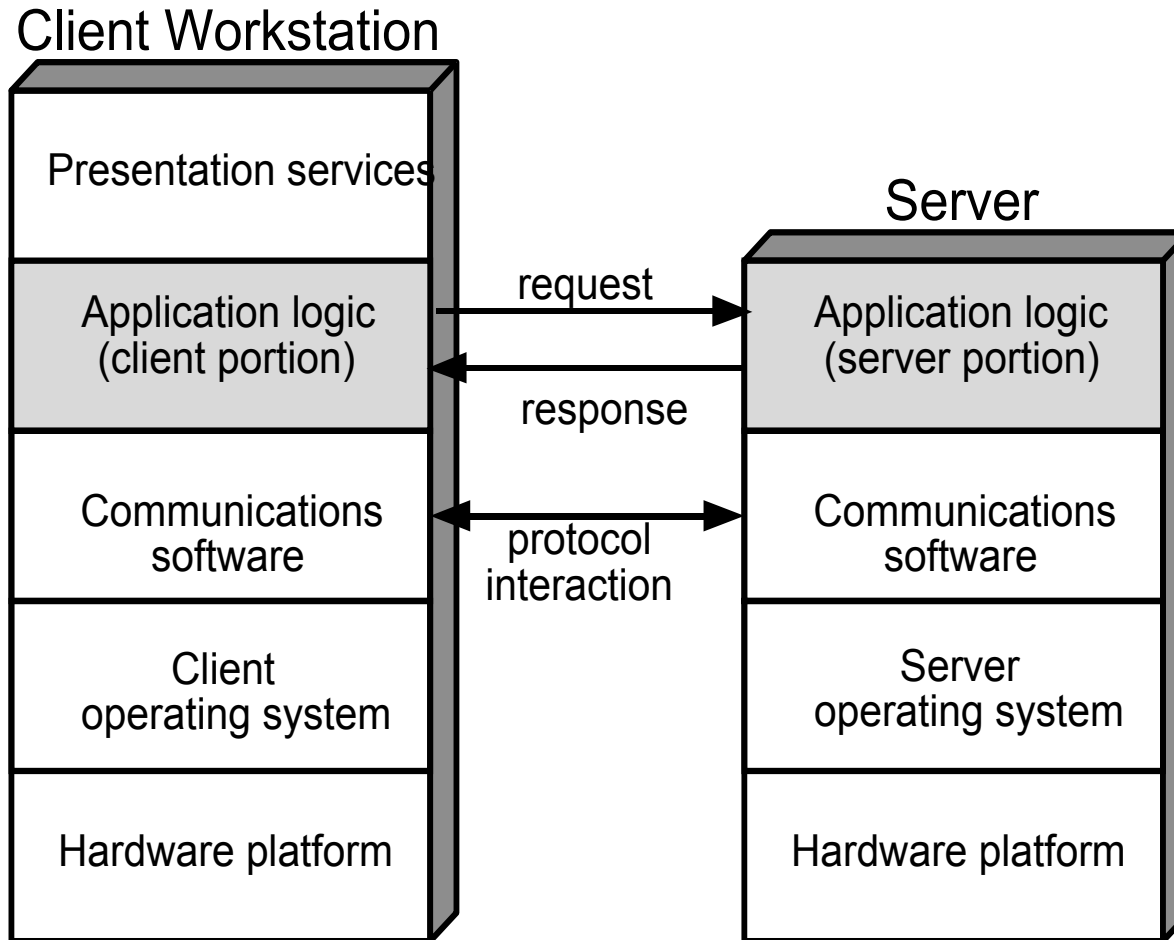
## 9.2 Middleware

- Ziele einer Middleware
  - Einführung einer zusätzlichen Schicht zwischen Betriebssystem und Anwendung, um höhere Abstraktionsebene zur Unterstützung verteilter Anwendungen zu erhalten.
  - Lokale Ressourcen einzelner Knoten sollen weiterhin vom (lokalen) Betriebssystem verwaltet werden.
  
- Mögliche Modelle (Paradigmen) für Middleware
  - Modell „Datei“: Behandlung aller lokaler und entfernter Ressourcen als Datei (Beispiele: Unix, Plan9)
  - Modell „Prozeduraufruf“: lokaler und entfernter Prozeduraufruf (Beispiel: RPC)
  - Modell „verteilte Objekte“: Objekte können lokal oder auf transparente Weise entfernt aufgerufen werden (Beispiel: RMI)
  - Modell „verteilte Dokumente“: Dokumente mit verweisen, wobei Ort des Dokuments transparent ist (Beispiel: WWW)



# Die Rolle von Middleware

- Client/Server-Kommunikation **ohne** Middleware

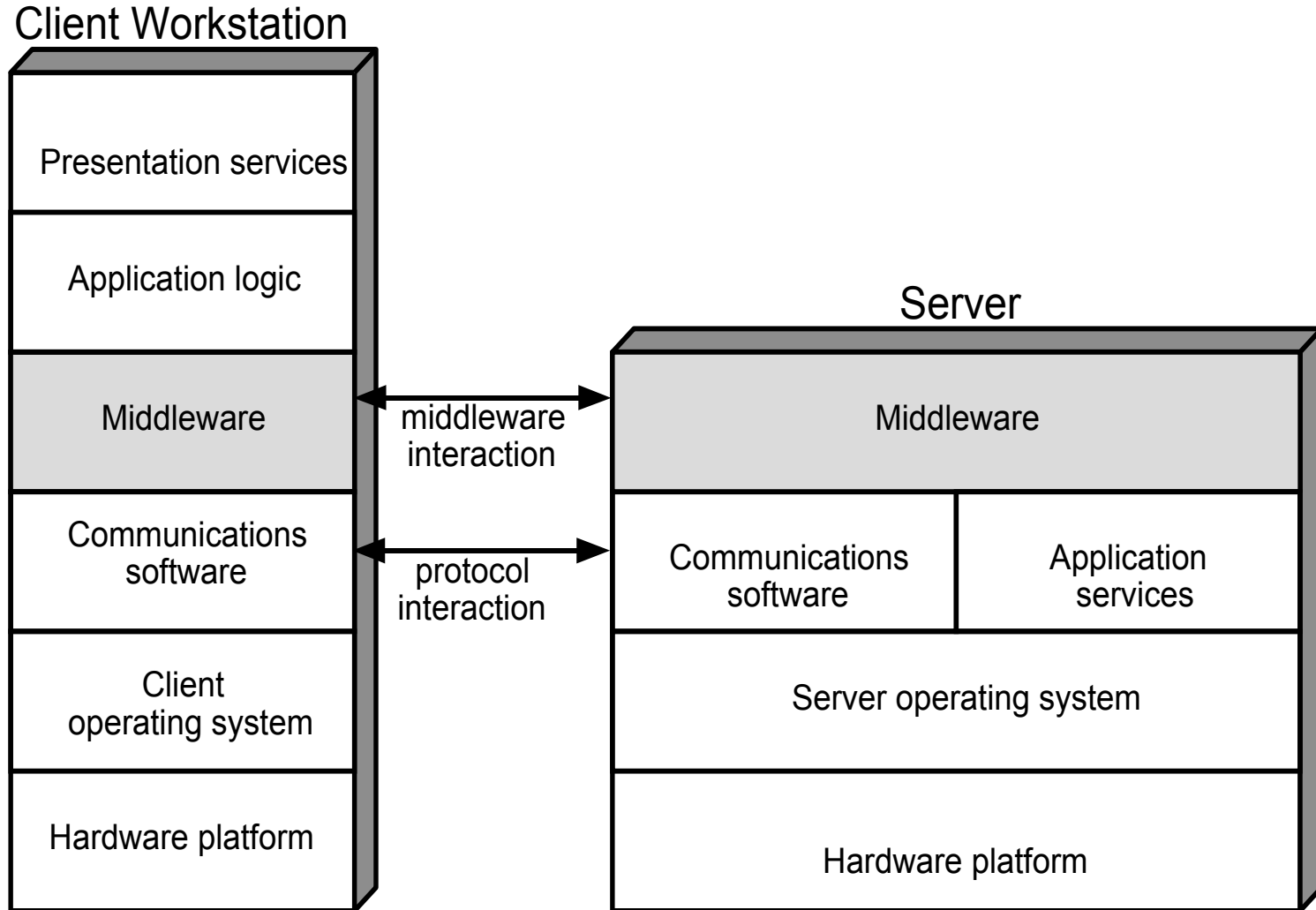


Quelle: „Operating Systems“, Stallings, Abb.13-7



# Die Rolle von Middleware

- Client/Server-Kommunikation **mit** Middleware

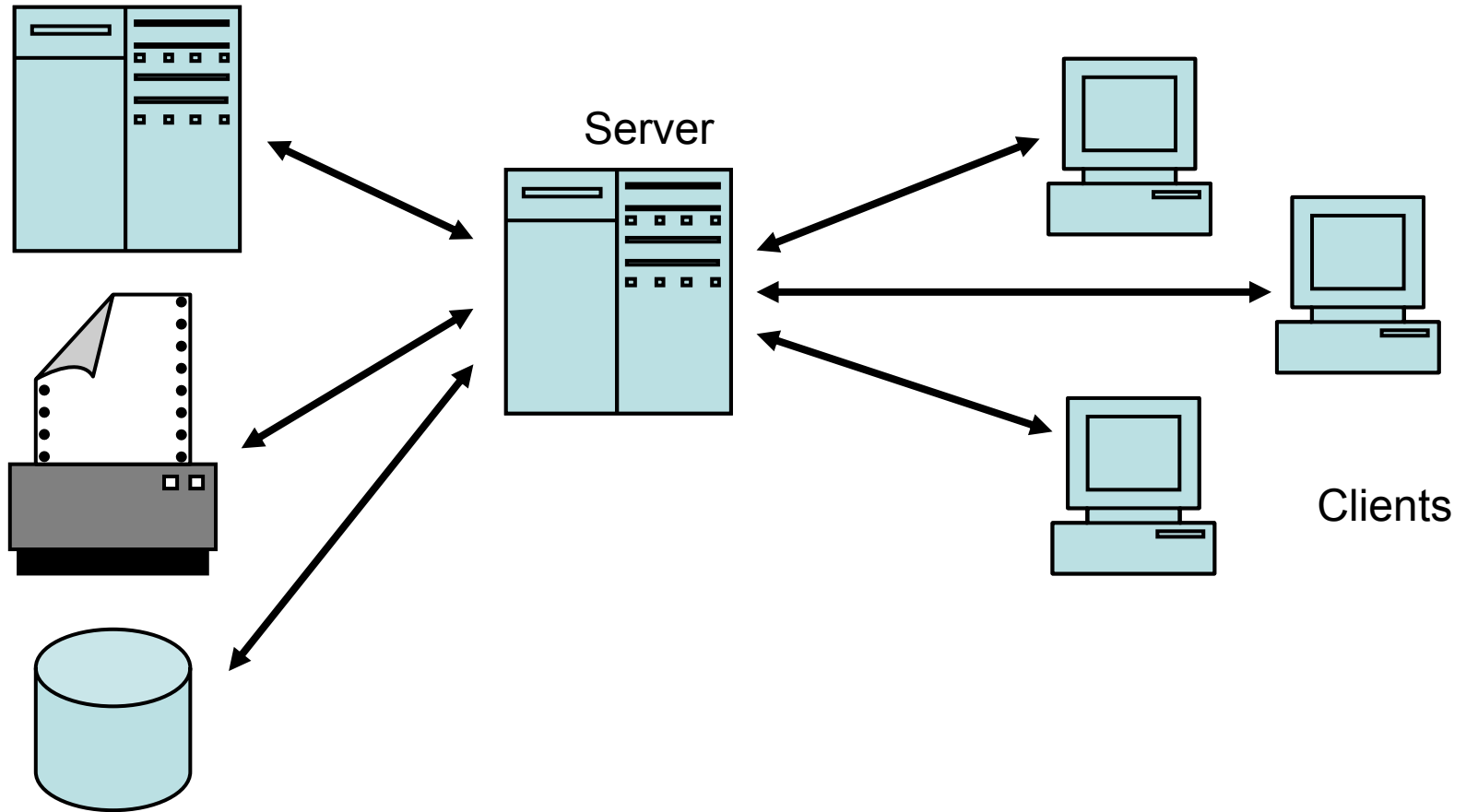


Quelle: „Operating Systems“, Stallings, Abb.13-12



# Three-Tier-Modell

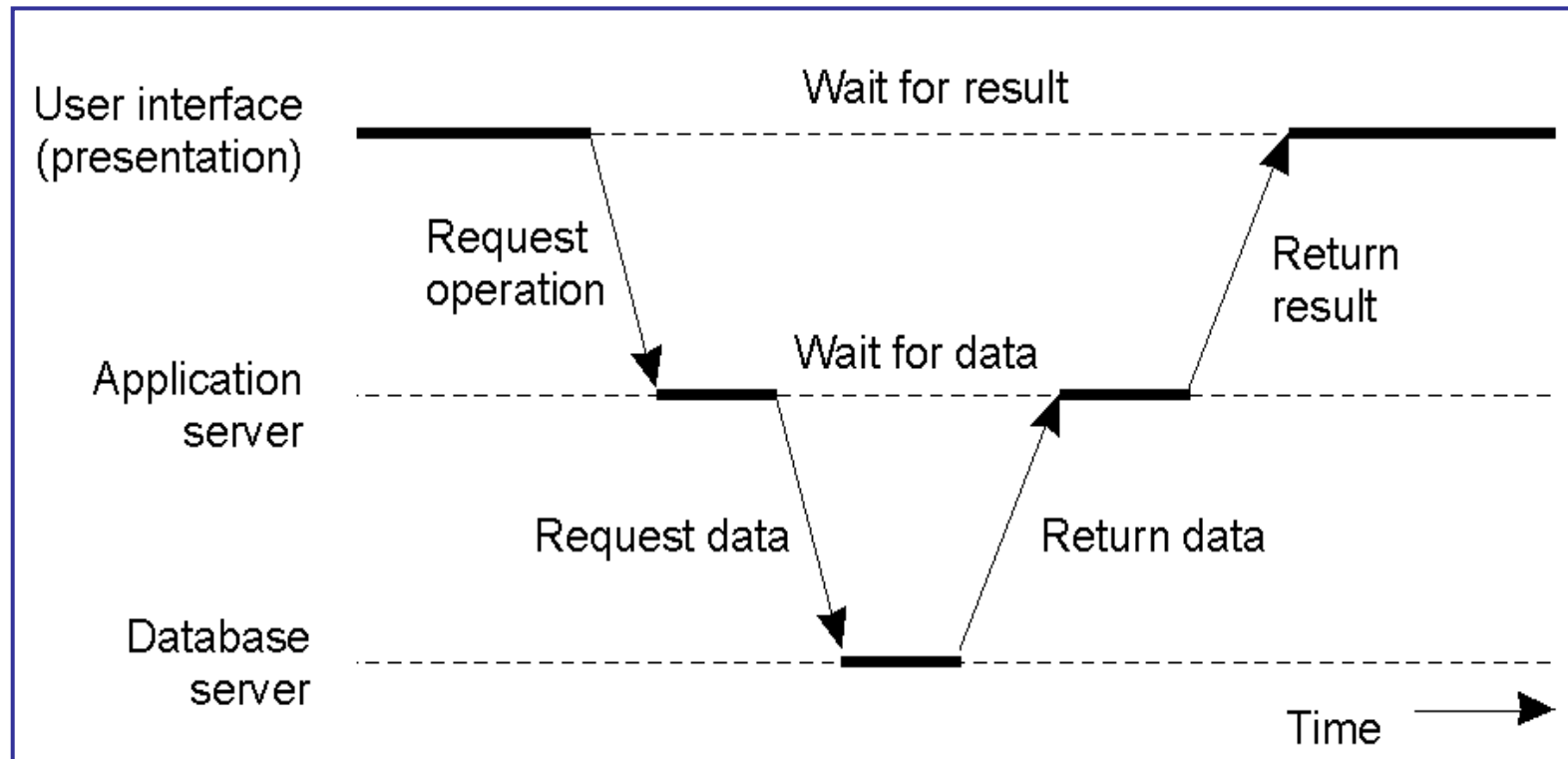
- ❑ Server selbst ist auch Client





# Three-Tier-Modell

- Erweitertes request-reply Verhalten

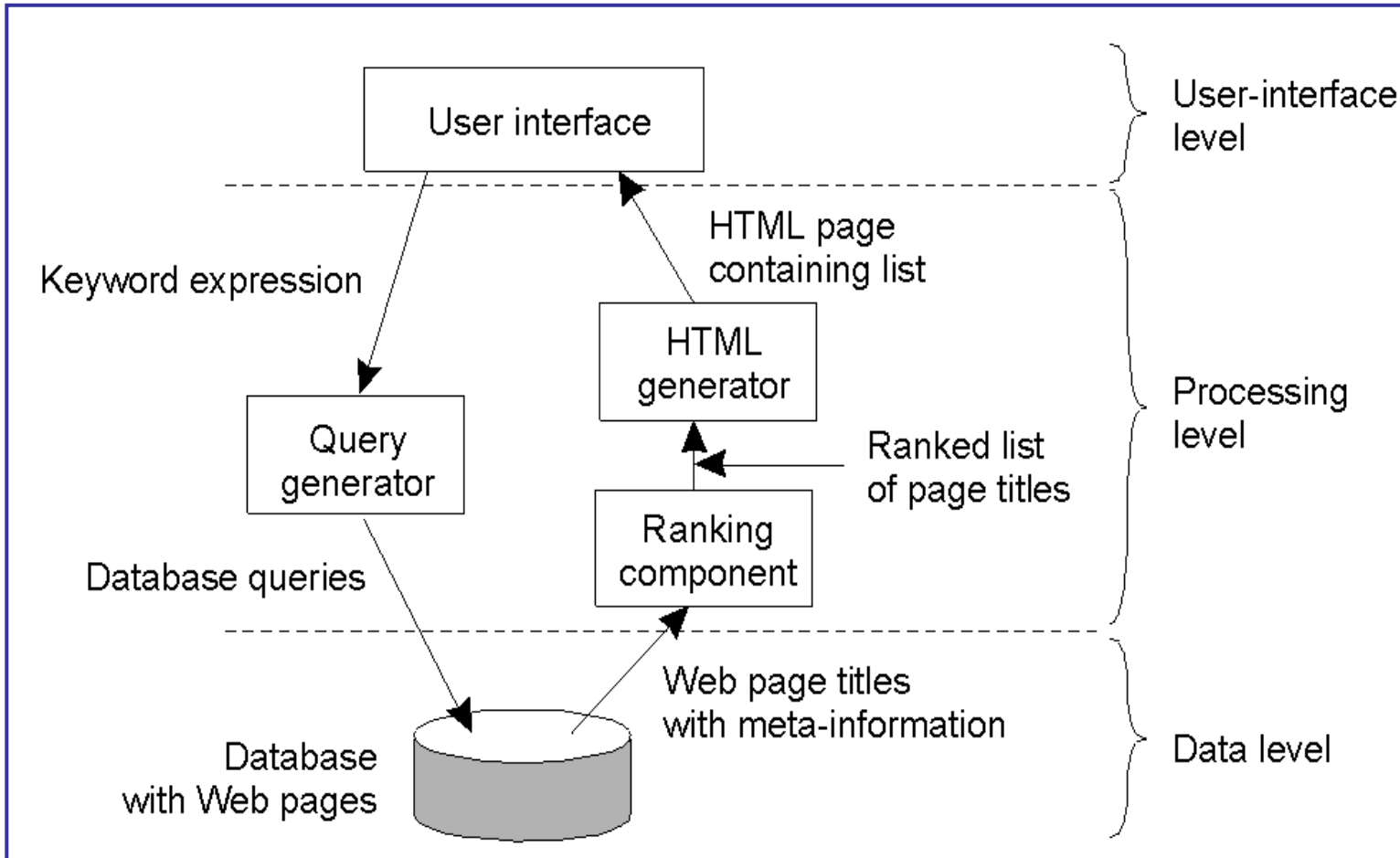


Quelle: „Distributed Systems“, Tanenbaum, van Steen, Abb.1-30



# 3+ -Schichten Client/Server-Modell

- Zwischenschichten in der Mitte, z.B. Web-Schicht



Quelle:  
„Distributed  
Systems“,  
Tanenbaum,  
van Steen,  
Abb.1-28



























































































































































