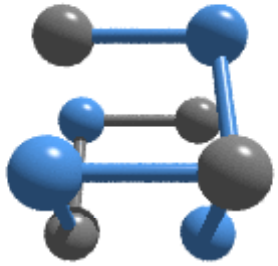


Vorlesung Softwaretechnik
- Datenbanken / Web-Architekturen -

Prof. Dr.-Ing. habil. Klaus-Peter Fähnrich

Wintersemester 2009/2010

Überblick LE 25 und LE 26

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

LE 25/26 Datenbanken

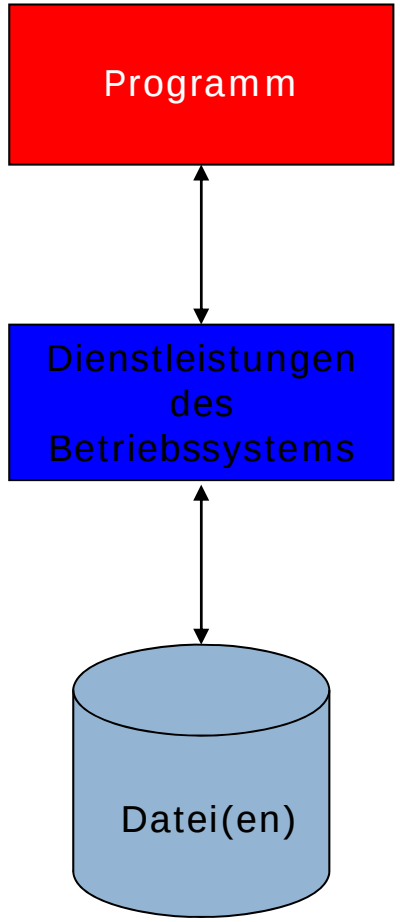
- Von Dateien zu Datenbanken
- Relationale Datenbanksysteme
- Objektorientierte Datenbanksysteme

LE 30 Web-Architekturen

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

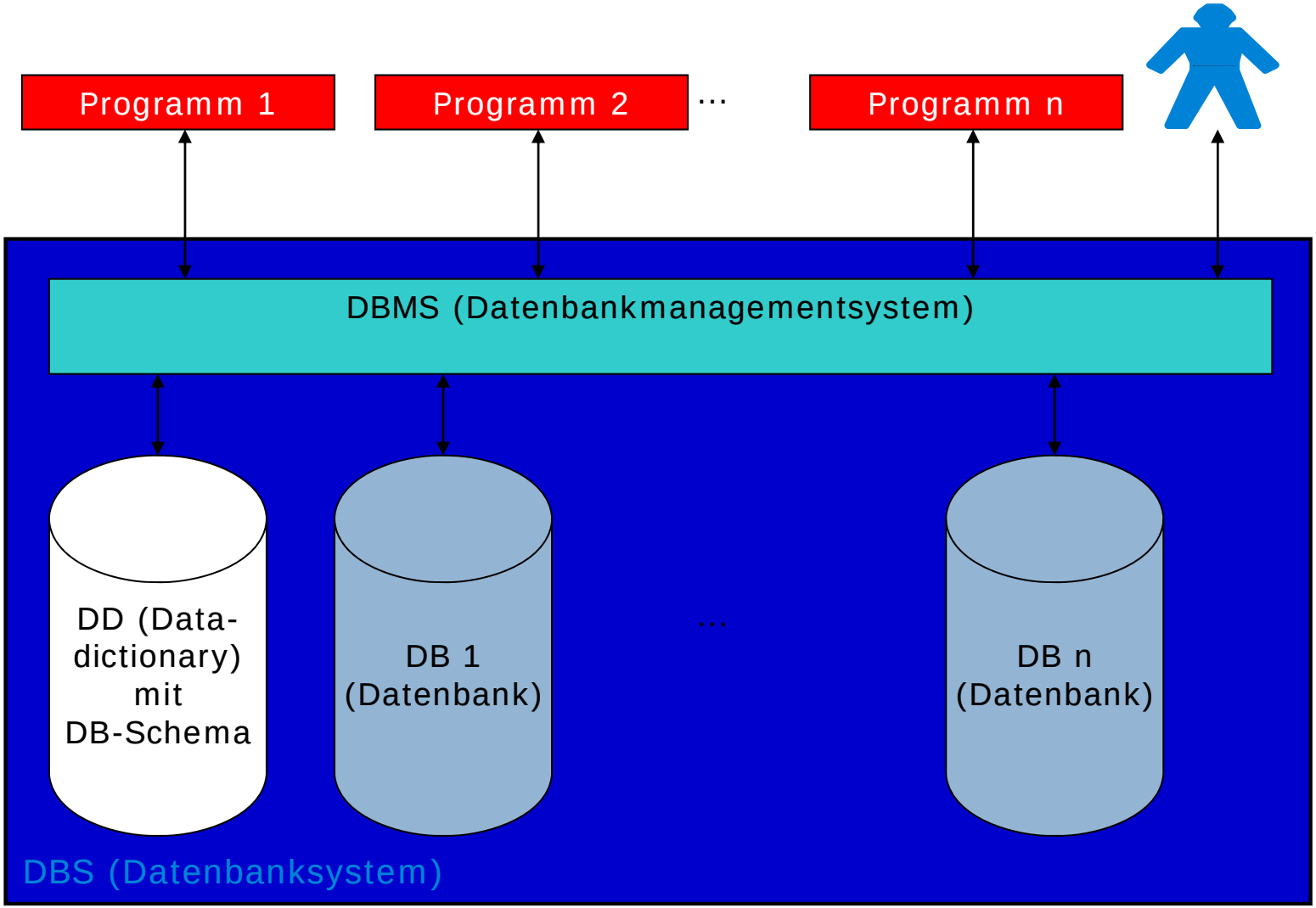
Von Dateien zu Datenbanken

- Persistente Datenhaltung in Dateien
- Anforderungen aus der Unternehmenspraxis
  - Integrierte Datenverwaltung innerhalb eines Anwendungsbereichs oder Unternehmens
  - Datenbeschreibungen unabhängig von den Programmen speichern.
- Ein Datenbanksystem (DBS) sorgt für die...
  - dauerhafte (persistente)
  - zuverlässige und
  - unabhängige Verwaltung sowie die
  - komfortable
  - flexible und
  - geschützte Verwendung
  - großer
  - integrierter und
  - mehrfachbenutzbarer Datenbanken.



Konzept eines Datenbanksystems

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Eigenschaften von DBS

- DBS (Datenbanksystem) = DBMS (Datenbankmanagementsystem)
  + DD (Data Dictionary)
  + nDB (Datenbank n >= 1)

- Eigenschaften
  - Zuverlässige Verwaltung
    - Konsistenz, Integrität, Unversehrtheit der Daten
    - Wiederanlauf des DBS
  - Unabhängige Verwaltung
    - Programme und DBS weitgehend unabhängig
    - Daten werden im DBS einheitlich beschrieben
  - Komfortable Verwendung
    - Kommunikation über abstrakte Schnittstelle
  - Flexible Verwendung
    - ad hoc-Zugriff über spezielle Anfragesprachen.
  - Geschützte Verwendung
    - Datenschutz vor unberechtigtem Zugriff
  - Große Datenbank
    - Nicht vollständig im Arbeitsspeicher

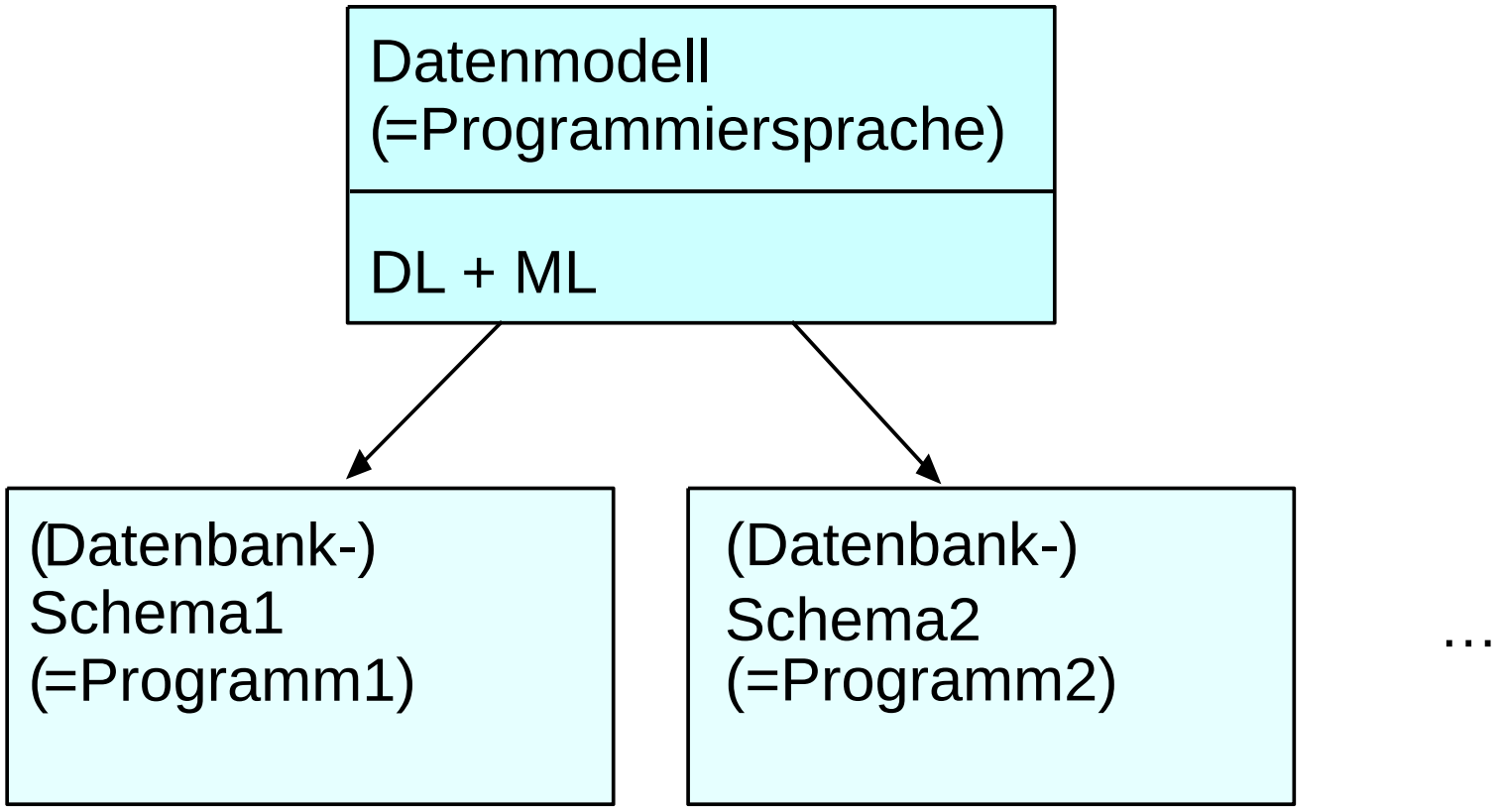
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Eigenschaften von DBS (2)

  - Integrierte Datenbank
    - Alle Daten redundanzarm gespeichert
    - Sichten (views) können definiert werden
    - Leistungsfähige Auswahlmechanismen, standardisierte Suchverfahren, effiziente Speicherungsstrategien für große Datenmengen
  - Mehrfachbenutzbare Datenbank
    - Nutzung durch mehrere Programme/Benutzer, u.U. gleichzeitig.
- Datenmodell legt fest...
  - durch welche Eigenschaften Datenelemente charakterisiert werden können
  - wie die Struktur der Datenelemente aussehen kann
  - welche Konsistenzbedingungen einzuhalten sind
  - welche Operationen zum Speichern, Auffinden, Ändern und Löschen von Daten erlaubt sind
- Syntax und Semantik eines Datenmodells
  - Definitionssprache (DL)
  - Manipulationssprache (ML)
- (Datenbank-) Schema
  - Beschreibt eine konkrete Datenbank, d.h. das Datenmodell wird auf einen Einsatzfall angewandt.

Vom Datenmodell zum Datenbankschema

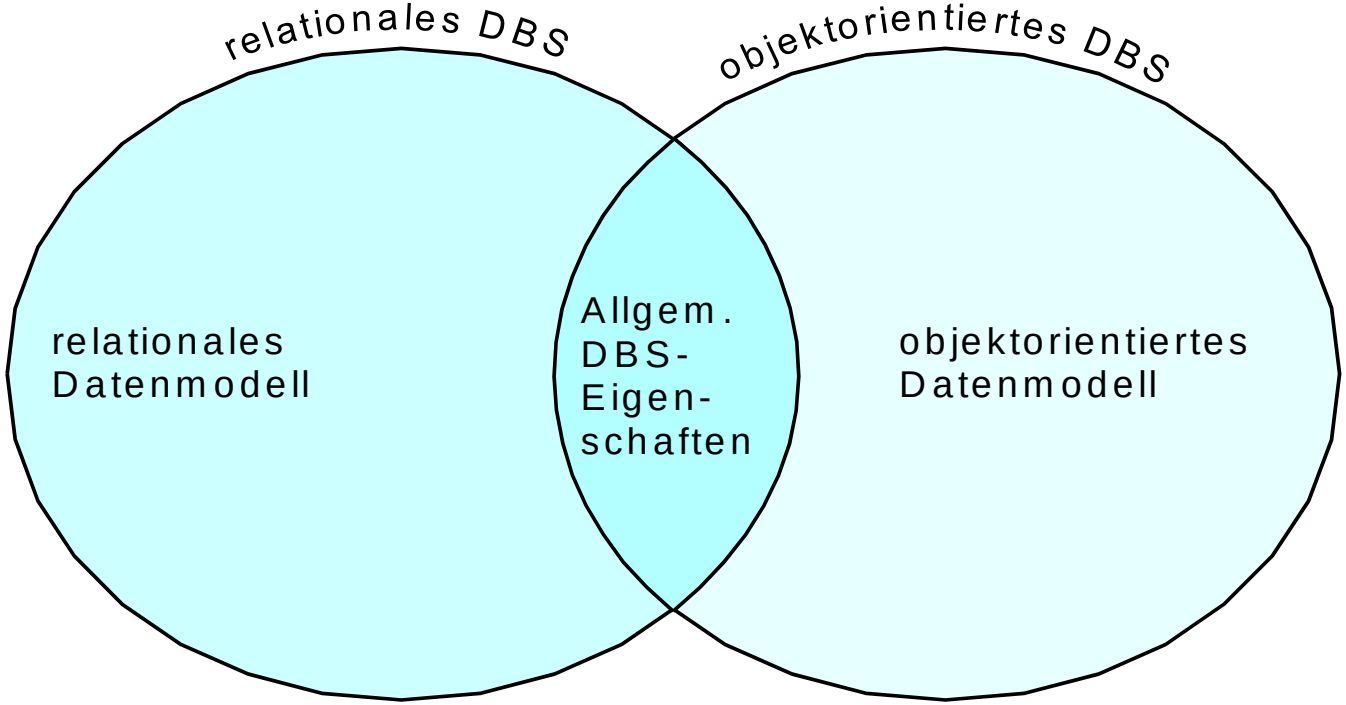
UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Relationale und Objektorientierte DBS

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

- Relationales Datenbanksystem (RDBS) : DBS liegt relationales Datenmodell zugrunde
- Objektorientiertes Datenbanksystem (ODBS) : DBS liegt objektorientiertes Datenmodell zugrunde



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Relationales Datenmodell

- Darstellung von Entitäten, ihren Eigenschaften und Beziehungen untereinander in Relationen.
- Jede Relation kann als Tabelle dargestellt werden. Die Spalten tragen die Namen der Attribute. In den Zeilen sind die Elemente (Tupel) aufgeführt.

Tabellenname — Schlüsselattribut (unterstrichen)

Firma	Kurzname	Name	Adresse	Kurzmitteilung
	Softtech	Softtech GmbH	Bochum	–
	Innosoft	Innovation & Software	Dortmund	Beiliegend erhalten Sie unseren neuesten…
	…	…	…	…

Tabelle bzw. Schema

← Attribute
← Tupel (Inhalt)

Objektorientiertes Datenmodell

- Objekte werden unverändert gespeichert. Sie werden nicht in Tabellen transformiert
- Schema-Definitionssprache
  - Objekt-Definitionssprache ODL (Object Definition Language) der ODMG (ODMG = Object Database Management Group)
- Jede Klasse des OOA-Modells wird in ODL durch eine Schnittstellendeklaration (interface declaration) beschrieben.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Vergleich RDBS und ODBS (1)

Relationales DBS	Objektorientiertes DBS
• Wertidentität (wertbasiert) o Identifikation über Schlüssel o Verweise über Fremdschlüssel und zusätzliche Tabellen • einfache Objekte o Attributtypen = elementare Typen o fest definiert • Sichtbare Attribute ▪ strikte Trennung zwischen Daten-strukturen (Schemata) und Anwendungs-operationen ▪ wenige generische Operationen • Keine Anbindung an Programmiersprachen ▪ Eigenes Typsystem ▪ Erweiterte deklarative Sprache oder ▪ Deklarative Sprache in prozedurale einbetten °Namensverknüpfung °Datenkonversion °keine Typüberprüfung ▪ »impedance mismatch« zwischen den Sprachen • Serverorientiert ▪ Zentralisierung von Datenbanken auf Servern	• Objektidentität (OID) o Jedes Objekt hat Identität, unabhängig von seinen Attributwerten o Verweise über OIDs • komplexe Objekte o Attributtypen = beliebige Typen o Typkonstruktoren struct, Set, Bag, List, Array o benutzerdefinierbar. • bedingt gekapselte Attribute ▪ Zugriff nur über bereitgestellte Operationen ▪ keine Kapselung für lesende Zugriffe. • Enge Anbindung an Programmiersprachen ▪ Typsystem Programmiersprache – Datenbank integriert ▪ DB-Schema kann in Programmiersprache beschrieben werden ▪ Spracherweiterung um OML (object manipulation language) ▪ Anbindung an OQL (object query language) • Clientorientiert ▪ Verteilung von Objekten auf vernetzten Clients.

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Vergleich RDBS und ODBS (2)

Relationales DBS	Objektorientiertes DBS
• Datenspeicherung ▪ Attributwerte werden gespeichert • persistent ▪ In Programmen deklarierte Daten sind transient • Schema-Definitionssprache (DL) ▪ DDL-Teil von SQL • Manipulationssprache (ML) ▪ DML-Teil von SQL ▪ eingebettetes SQL • externe Schemata ▪ Definierbar ▪ In DD abgelegt • Eingabe von Daten ▪ Mit insert • Semantik der Anwendung ▪ Verschwindet (nur Tabellen und Fremdschlüssel)	• Objektspeicherung ▪ Attributwerte & Operationen werden gespeichert – heute in der Regel nur Attribute • persistent & transient ▪ Objekte können persistent oder transient sein • Schema-Definitionssprache (DL) ▪ ODL oder PL-ODL • Manipulationssprache (ML) ▪ OQL ▪ OML in Java, C++ bzw. Smalltalk. • externe Schemata ▪ Nicht definierbar, nur als Programme ▪ Nicht in DD abgelegt • Eingabe von Daten ▪ Nur über Anwendungs-programm, nicht mit OQL • Semantik der Anwendung ▪ Bleibt erhalten (durch explizite Definition von Relationen).

Überblick LE 30

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

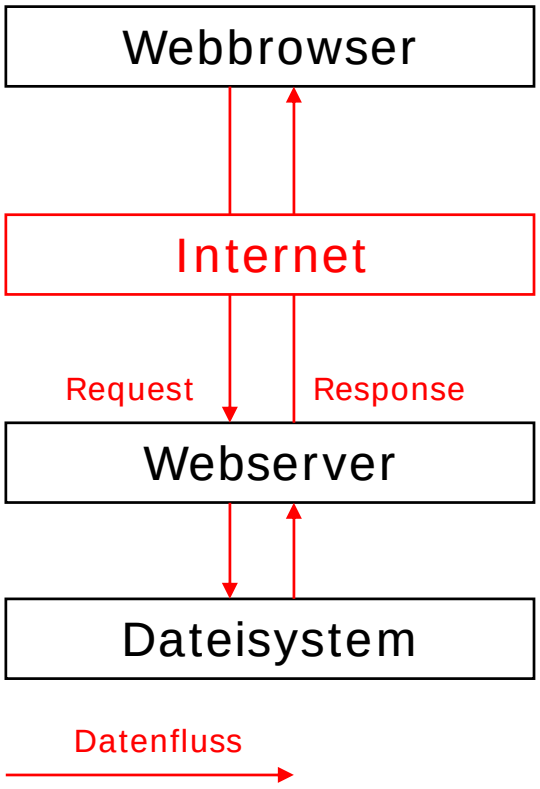
LE 25/26 Datenbanken

LE 30 Web-Architekturen

1. Statische und dynamische Web-Inhalte
2. Webbrowser
3. Webserver

Statische Web-Inhalte

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme



Anfrage an den Webserver

- Hyperlink im HTML-Dokument
- Direkter Aufruf der Webseite

Request-/Response-Zyklus (HTTP-Protokoll)

- Aufbau einer TCP-Verbindung
- HTTP-Request an den in der URL spezifizierten Web-Dämon
- Zurücksenden der angeforderten Seite über die gleiche Verbindung
- Schließen der Verbindung bzw. Halten der Verbindung falls weitere Daten wie z.B. Grafiken folgen.

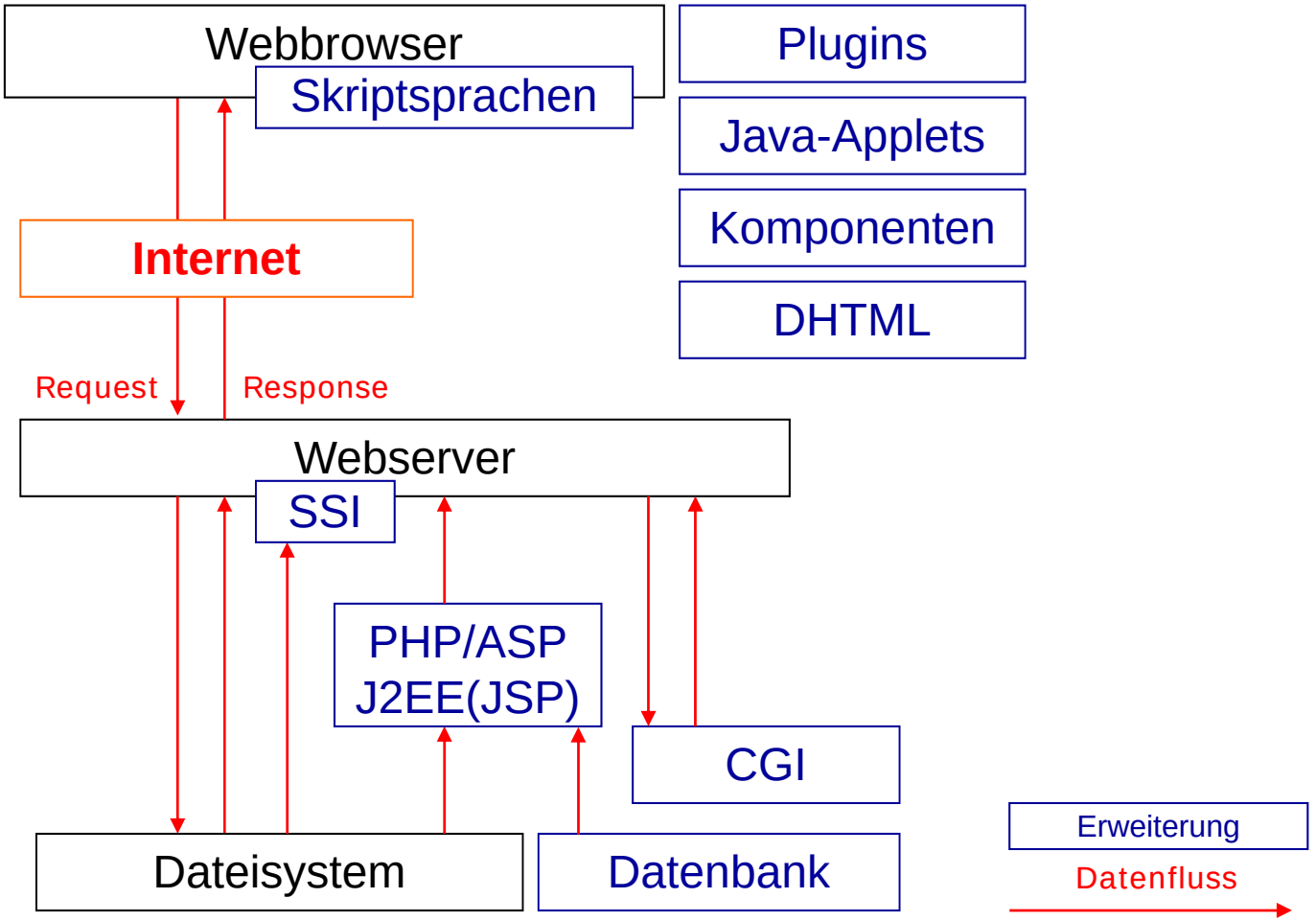
Darstellung der Information

- Interpretation und Darstellung des HTML-Dokuments

Dynamische Web-Inhalte

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Um Webinhalte dynamisch darzustellen ist es notwendig, den Browser bzw. den Server zu erweitern.



UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Aufgaben dynamischer Webinhalte

Webbrowser (Client)

- Steuerung von Inhalt und Aussehen von Dokumenten
- Steuerung des Browserverhaltens
- Interaktion mit dem Inhalt des Dokuments
- Interaktion mit dem Benutzer
- Zustandshaltung
- Interaktion mit Applets
- Interaktion mit Komponenten

Webserver

- Zugriff auf Datenbanken und Aufbereitung der Daten
- Auswertung von Formularen und Ausführung von Applikationen
  - Zähler, Gästebücher, Communities, Lernumgebungen etc.
  - Onlinebanking, Shopping-Systeme
- Generierung von HTML-Code angepasst an den Client
  - Sprache, Browsertyp
- Zustandshaltung

Webbrowser (Mozilla)

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Die Mozilla Suite besteht aus folgenden Komponenten:

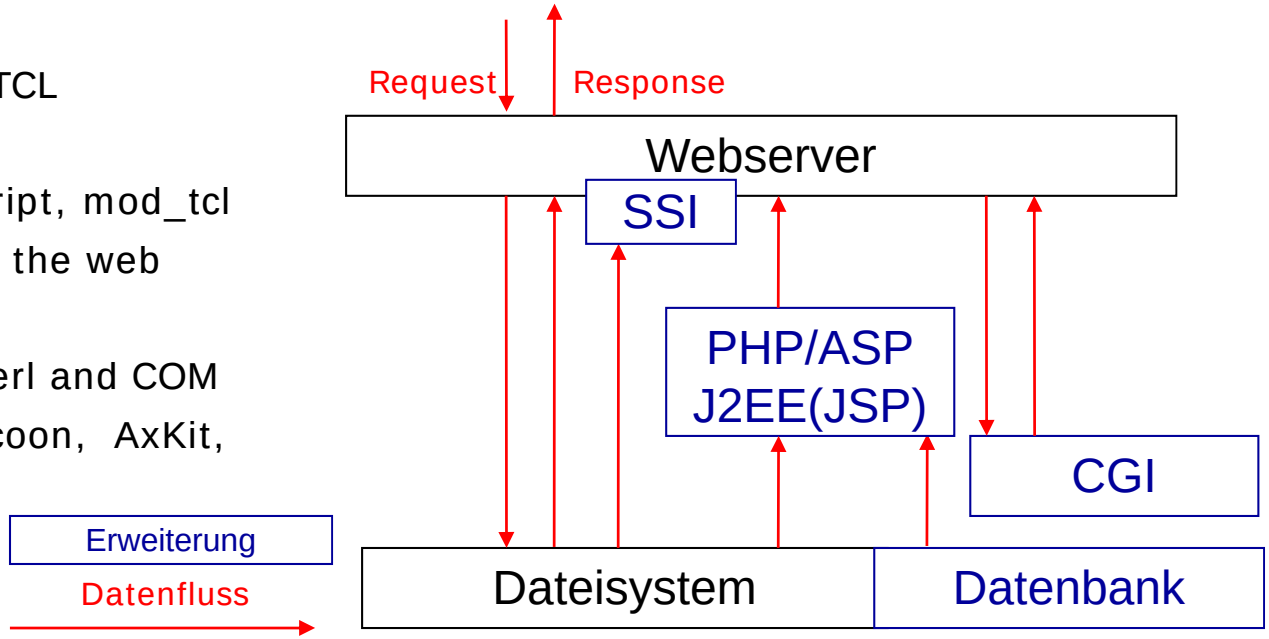
- Navigator (Webbrowser)
- Cookie Manager (Anzeigen und verwalten aller gespeicherten Cookies)
- Password Manager (Merkt sich Passwörter für geschützte Webseiten)
- Form Manager (Merkt sich Formulareingaben und füllt Formulare automatisch aus)
- Mail & News (eMail und News Client)
- Adressbuch (Adressverwaltung für Mail und News)
- Composer (WYSIWYG HTML Editor)
- Chatzilla (IRC Client)
- Themes (Mozilla's Aussehen kann mit Themes komplett verändert werden)
- Unterstützte Standards:
  - HTML 4.01 / XHTML 1.0
  - CSS 1 / CSS 2 teilweise
  - PNG Grafiken inkl. Alpha Transparenz
  - XML 1.0
  - ECMA Script
  - Java Applets, wenn das Java Runtime Environment installiert ist
  - DOM 1.0

UNIVERSITÄT LEIPZIG
Institut für Informatik
Betriebliche Informationssysteme

Webserver (Apache Project)

Apache Project (http://www.apache.org)

- **HTTP Server**
  - Sub-Projects: Docs, Test
- **The Apache Portable Runtime**
- **Jakarta Server-side Java**
  - Sub-Projects: Alexandria, Ant, Avalon, BCEL, Cactus, Commons, ECS, James, Jetspeed, JMeter, Log4J, Lucene, ORO, POI, Regexp, Slide, Struts, Taglibs, **Tomcat**, Turbine, Velocity, Watchdog
- **Dynamic websites using Perl**
- **PHP Server-side**
- Dynamic websites using TCL
  - Sub-Projects: mod_dtcl, neowebscript, mod_tcl
- XML solutions focused on the web
  - Sub-Projects: Xerces, C++ (with Perl and COM bindings), Xalan, Cocoon, AxKit, FOP, Xang, SOAP, Batik, Crimson



Literatur

- Balzert H., Lehrbuch der Softwaretechnik, 2. Auflage, Spektrum Akademischer Verlag,Heidelberg, 2000.

- Codd, E. F., The Relational Model for Database Management, Version 2, Addison-Wesley, 1990
- Date C. J., An Introduction to Database Systems, Volume 1, 5. Edition, Addison-Wesley
- Vetter M., Aufbau betrieblicher Informationssysteme mittels konzeptioneller Datenmodellierung, Teubner Verlag, Stuttgart, 1987
- Vossen G., Datenmodelle, Datenbanksprachen und Datenbank-managementsysteme, 4. Auflage, Oldenbourg Verlag, 2000
- Bensber F., Dewanto L., Klein M., Quo vadis, Datenbank?, in Java Magazin, April 2000, S. 38-53, Software & Support Verlag
- Cattell R.G.G., Barry D. K., The Object Data Standard: ODMG 3.0, Morgan Kaufmann Publishers, 2000