

Vorlesung Softwaretechnik - Datenbanken / Web-Architekturen -

Prof. Dr.-Ing. Klaus-Peter Fährnich
WS 2007/2008

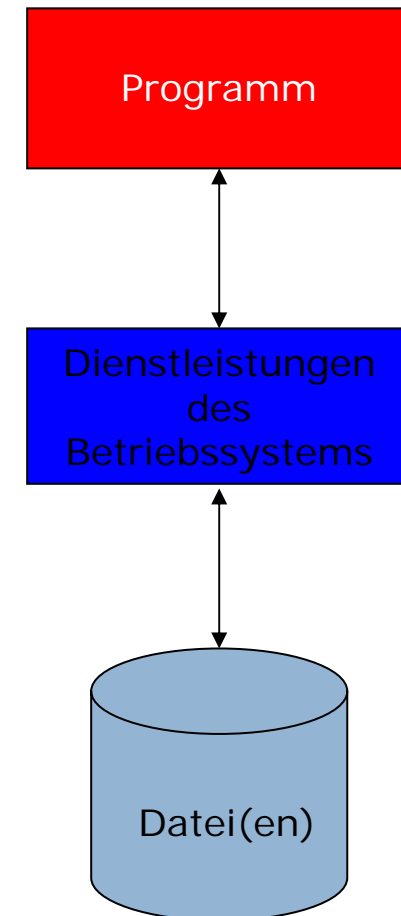
LE 25/26 Datenbanken

- Von Dateien zu Datenbanken
- Relationale Datenbanksysteme
- Objektorientierte Datenbanksysteme

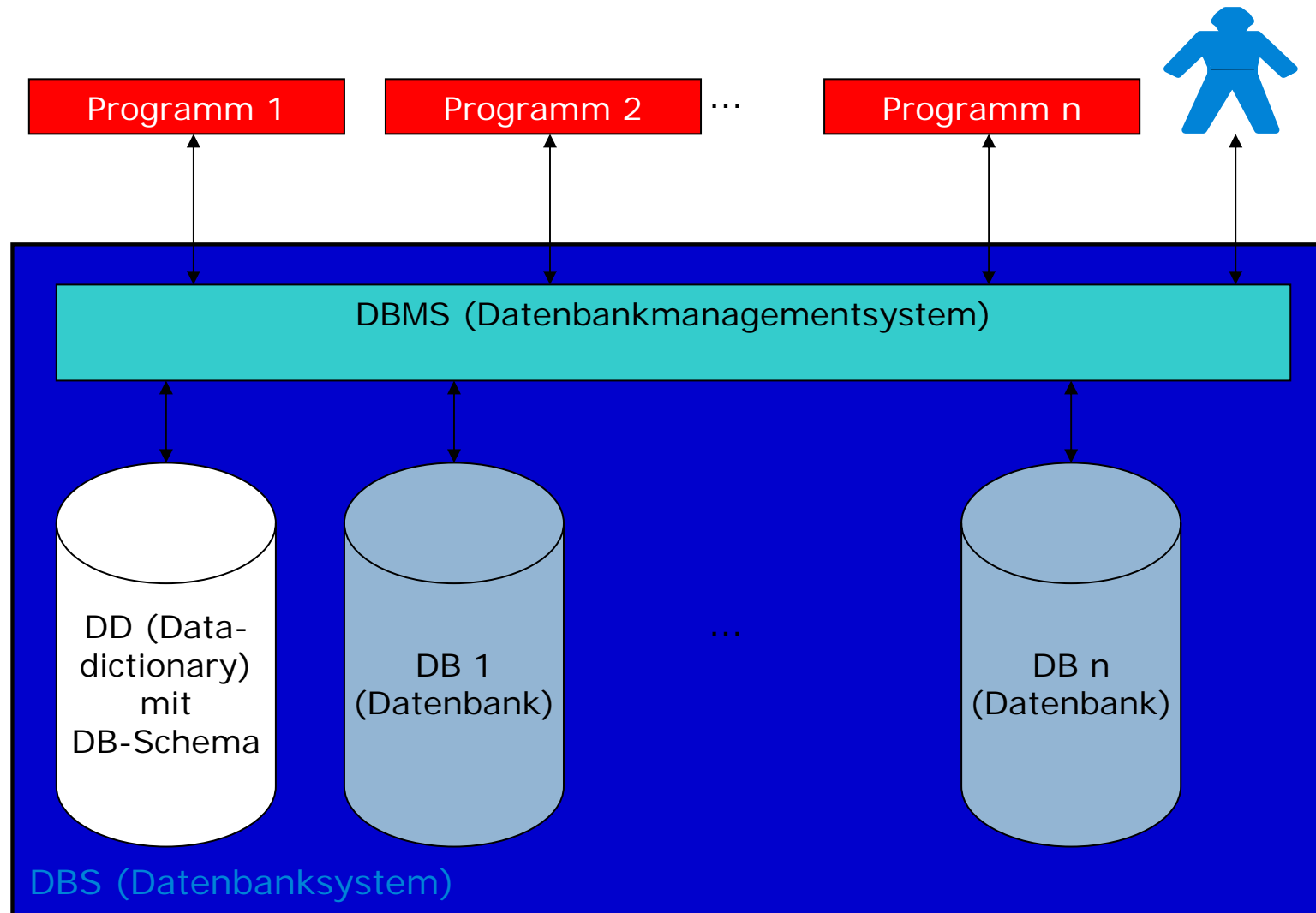
LE 30 Web-Architekturen

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



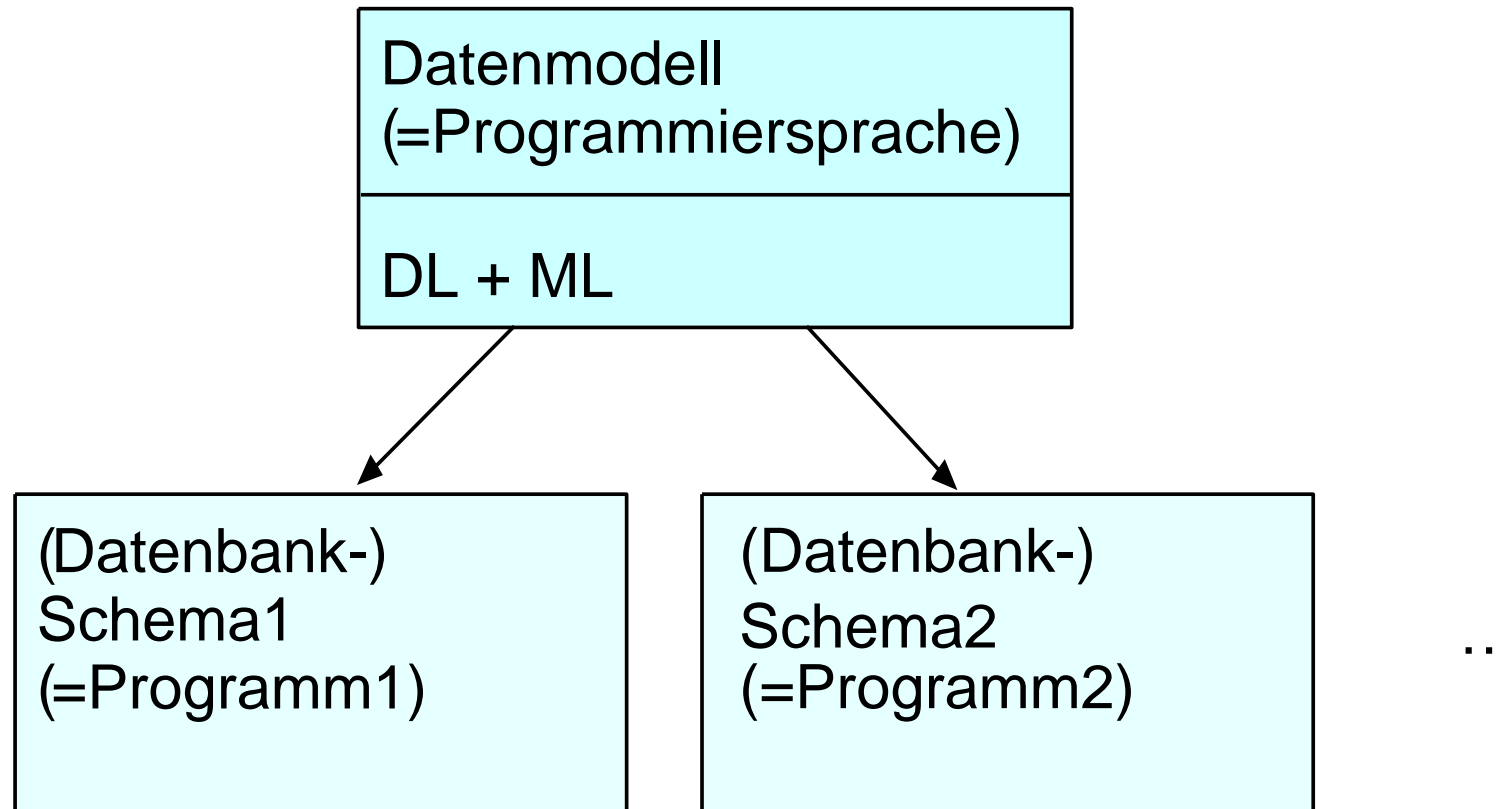
Eigenschaften von DBS

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

- 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

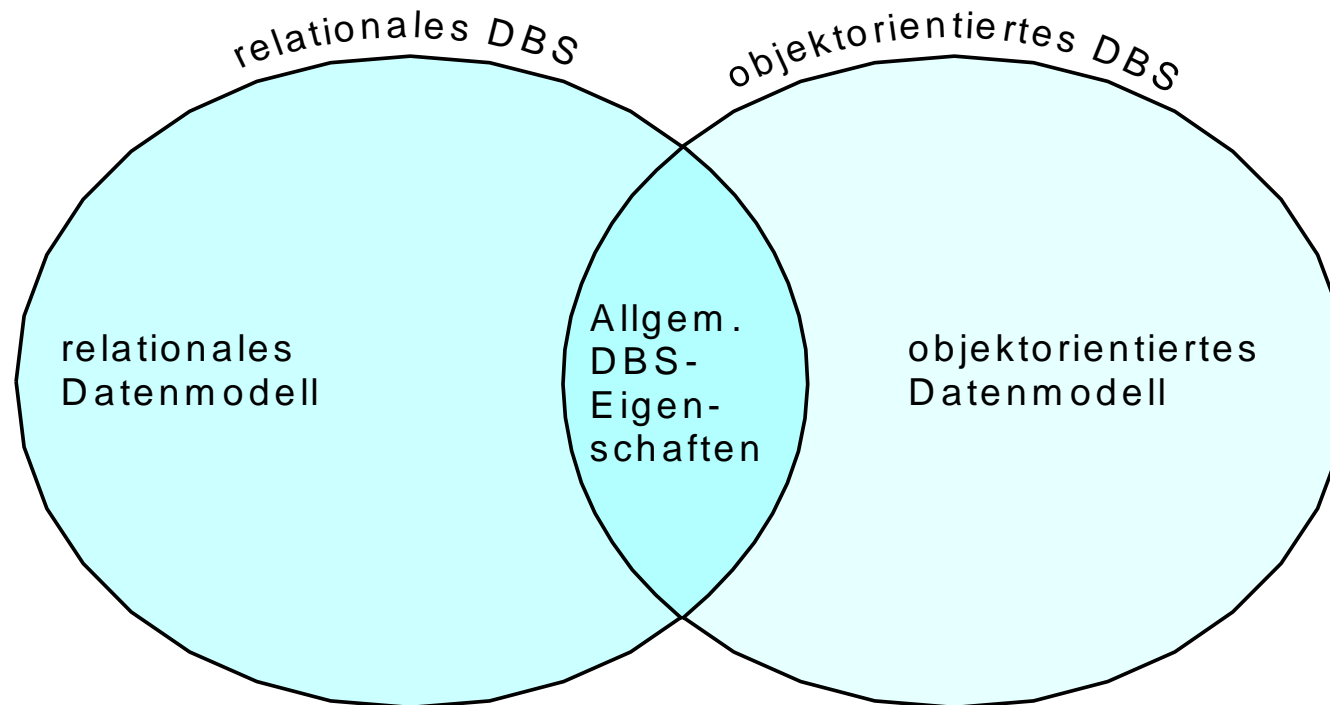
Eigenschaften von DBS (2)

- 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.



Relationale und Objektorientierte DBS

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



Datenmodelle

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	<u>Kurzname</u>	Name	Adresse	Kurzmitteilung
Softtech	Softtech GmbH	Bochum	–	
Innosoft	Innovation & Software	Dortmund	Beiliegend erhalten Sie unseren neuesten...	
...

← Attribute
← Tupel (Inhalt)

Tabelle bzw. Schema

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.

Vergleich RDBS und ODBS (1)

Relationales DBS	Objektorientiertes DBS
<ul style="list-style-type: none"> • Wertidentität (wertbasiert) <ul style="list-style-type: none"> o Identifikation über Schlüssel o Verweise über Fremdschlüssel und zusätzliche Tabellen • einfache Objekte <ul style="list-style-type: none"> o Attributtypen = elementare Typen o fest definiert • Sichtbare Attribute <ul style="list-style-type: none"> ▪ strikte Trennung zwischen Datenstrukturen (Schemata) und Anwendungsoperationen ▪ wenige generische Operationen • Keine Anbindung an Programmiersprachen <ul style="list-style-type: none"> ▪ Eigenes Typsystem ▪ Erweiterte deklarative Sprache oder ▪ Deklarative Sprache in prozedurale einbetten <ul style="list-style-type: none"> ° Namensverknüpfung ° Datenkonversion ° keine Typüberprüfung ▪ »impedance mismatch« zwischen den Sprachen • Serverorientiert <ul style="list-style-type: none"> ▪ Zentralisierung von Datenbanken auf Servern 	<ul style="list-style-type: none"> • Objektidentität (OID) <ul style="list-style-type: none"> o Jedes Objekt hat Identität, unabhängig von seinen Attributwerten o Verweise über OIDs • komplexe Objekte <ul style="list-style-type: none"> o Attributtypen = beliebige Typen o Typkonstruktoren struct, Set, Bag, List, Array o benutzerdefinierbar. • bedingt gekapselte Attribute <ul style="list-style-type: none"> ▪ Zugriff nur über bereitgestellte Operationen ▪ keine Kapselung für lesende Zugriffe. • Enge Anbindung an Programmiersprachen <ul style="list-style-type: none"> ▪ Typsystem Programmiersprache – Datenbank integriert ▪ DB-Schema kann in Programmiersprache beschrieben werden ▪ Spracherweiterung um OML (object manipulation language) ▪ Anbindung an OQL (object query language) • Clientorientiert <ul style="list-style-type: none"> ▪ Verteilung von Objekten auf vernetzten Clients.

Vergleich RDBS und ODBS (2)

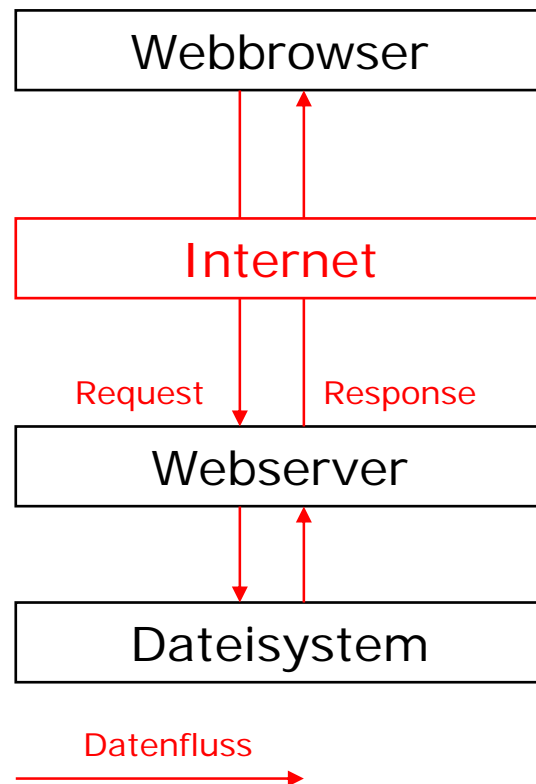
Relationales DBS	Objektorientiertes DBS
<ul style="list-style-type: none"> • Datenspeicherung <ul style="list-style-type: none"> ▪ Attributwerte werden gespeichert • persistent <ul style="list-style-type: none"> ▪ In Programmen deklarierte Daten sind transient • Schema-Definitionssprache (DL) <ul style="list-style-type: none"> ▪ DDL-Teil von SQL • Manipulationssprache (ML) <ul style="list-style-type: none"> ▪ DML-Teil von SQL ▪ eingebettetes SQL • externe Schemata <ul style="list-style-type: none"> ▪ Definierbar ▪ In DD abgelegt • Eingabe von Daten <ul style="list-style-type: none"> ▪ Mit insert • Semantik der Anwendung <ul style="list-style-type: none"> ▪ Verschwindet (nur Tabellen und Fremdschlüssel) 	<ul style="list-style-type: none"> • Objektspeicherung <ul style="list-style-type: none"> ▪ Attributwerte & Operationen werden gespeichert – heute in der Regel nur Attribute • persistent & transient <ul style="list-style-type: none"> ▪ Objekte können persistent oder transient sein • Schema-Definitionssprache (DL) <ul style="list-style-type: none"> ▪ ODL oder PL-ODL • Manipulationssprache (ML) <ul style="list-style-type: none"> ▪ OQL ▪ OML in Java, C++ bzw. Smalltalk. • externe Schemata <ul style="list-style-type: none"> ▪ Nicht definierbar, nur als Programme ▪ Nicht in DD abgelegt • Eingabe von Daten <ul style="list-style-type: none"> ▪ Nur über Anwendungs-programm, nicht mit OQL • Semantik der Anwendung <ul style="list-style-type: none"> ▪ Bleibt erhalten (durch explizite Definition von Relationen).

LE 25/26 Datenbanken

LE 30 Web-Architekturen

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

Statische Web-Inhalte



Anfrage an den Webserver

- Hyperlink im HTML-Dokument
- Direkter Aufruf der Webseite

Request-/Response-Zyklus (HTTP-Protokoll)

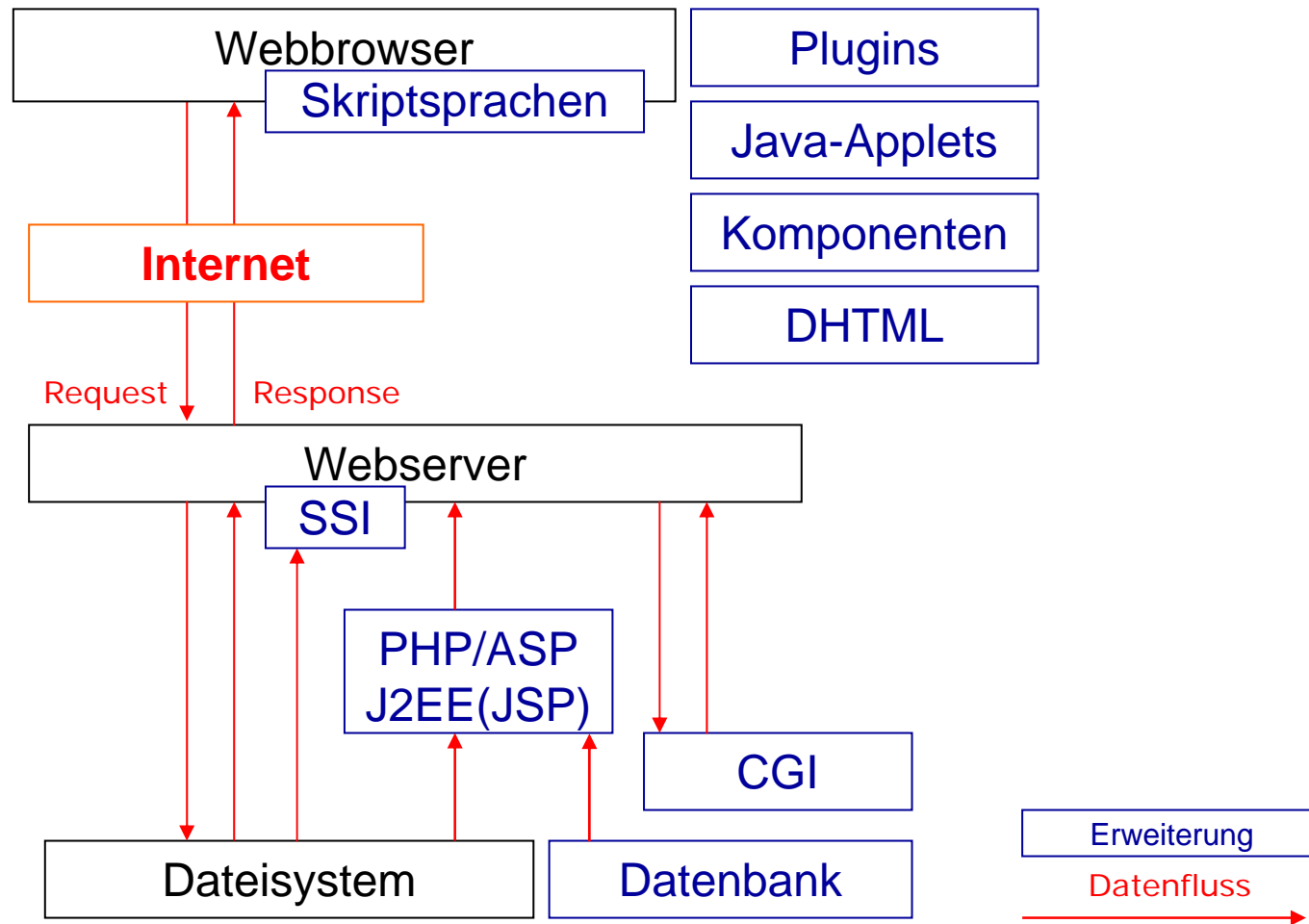
- Aufbau einer TCP-Verbindung
- HTTP-Request an den in der URL spezifizierten Web-Dämon
- Zurücksenden der angeforderten Seite über die gleiche Verbindung
- Schließen der Verbindung bzw. Halten der Verbindung falls weitere Daten wie z.B. Grafiken folgen.

Darstellung der Information

- Interpretation und Darstellung des HTML-Dokuments

Dynamische Web-Inhalte

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



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)

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

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

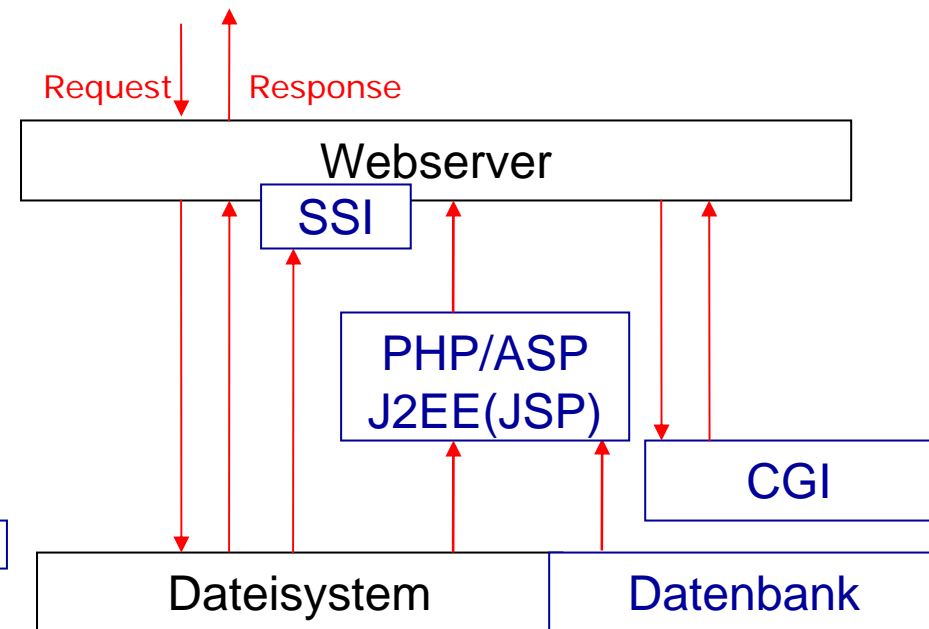
▪ PHP Server-side

▪ 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 Datenbankmanagementsysteme, 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