



Kommunikation und Datenhaltung

Kapitel 2: Architektur von Datenbanksystemen



Überblick über den Datenhaltungsteil

- Einleitung
 - Motivation und Grundlagen
 - Architektur von Datenbanksystemen
- Datenbankabfragen
 - Relationenmodell und Relationenalgebra
 - Relationale Datenbanksprachen (SQL)
- Datenbankentwurf
 - ER- und EER-Modell
 - Abbildung von ER-Modellen auf das Relationenmodell
 - Relationaler Entwurf
 - Sprachen zur Datenbankdefinition
- Transaktionsverwaltung
- Anfrageoptimierung
- Datenbankanwendungsentwicklung

Agenda – Architekturen von DBS

Grundlagen

Schema-
architektur

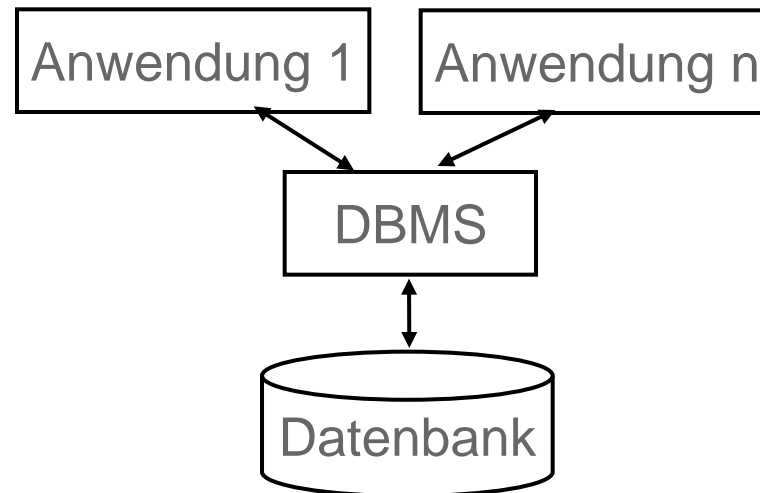
System-
architektur

Anwendungs-
architektur

- Letztes Mal: Vorteile eines DBMS (gegenüber einfacher Dateispeicherung)
- Heute: Prinzipien, Komponenten und Funktionen eines DBMS (Überblick über ein DBMS)
 1. Schemaarchitektur
(Unterscheidung: konzeptionelles, internes, externes Schema)
 2. Systemarchitektur
(Aufbau eines Datenbanksystems/ Komponenten)
 3. Anwendungsarchitektur
(Einbindung des DBS in die Applikation; Aufteilung der Funktionalität)

Terminologie

- DBMS: Datenbank-Management-System
 - Software zur Datenverwaltung
- DB: Datenbank
 - Strukturierter, von DBMS verwalteter Datenbestand
- DBS: Datenbanksystem
 - DBMS + Datenbank(en)



Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

Aufgaben eines DBMS

Basisfunktionalität eines DBMS – Codd hat Anforderungen
1982 in 9 Punkten zusammengefasst:

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

- Integration: einheitliche (→ Datenbankmodell), nichtredundante Datenverwaltung
- Operationen: Speichern, Suchen, Ändern
- Katalog (Data Dictionary): Zugriff auf Metadaten
- Benutzersichten
- Konsistenzüberwachung: Korrektheit der Datenbank
- Zugriffskontrolle: Ausschluss unautorisierter Zugriffe
- Transaktionen: mehrere DB-Operationen als Funktionseinheit
- Synchronisation: parallele Transaktionen koordinieren
- Datensicherung: Wiederherstellung von Daten nach Systemfehlern

Datenunabhängigkeit (1)

- Lange Lebensdauer von Datenbanken und Anwendungssystemen
 - Modifikationen/ Erweiterungen der Datenspeicherung sowie externer Schnittstellen während der Lebensdauer
- ⇒ Notwendigkeit der Entkopplung von Datenbank und Anwendungen
- Prinzip der „Datenunabhängigkeit“
- Intuitiv (formalere Darstellung folgt):
 - Möglichkeit, die Speicherung der Daten zu ändern, ohne dass dies nach außen sichtbar wird
 - ⇒ Abkopplung der Anwendung von Änderungen an der DB
 - Möglichkeit, „Anwendungsschnittstellen“ zu ändern, ohne dass sämtliche Anwendungen betroffen sind
 - ⇒ Abkopplung der DB von Änderungen an Anwendungen

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

Schemaarchitektur (1)

- Zur Unterstützung der Datenunabhängigkeit:
Drei-Ebenen Schemaarchitektur
(Grundprinzip moderner Datenbanksysteme)
- 3 Schemata:
 1. Konzeptuelles Schema
 - Ergebnis der Datenmodellierung, des Datenbankentwurfs und der Datendefinition
 2. Internes Schema
 - Datenorganisation und Zugriffspfade des konzeptuellen Schemas
 3. Externes Schema
 - Benutzersichten auf das konzeptuelle Schema

Anwendungsprogramme arbeiten auf externen Schemata

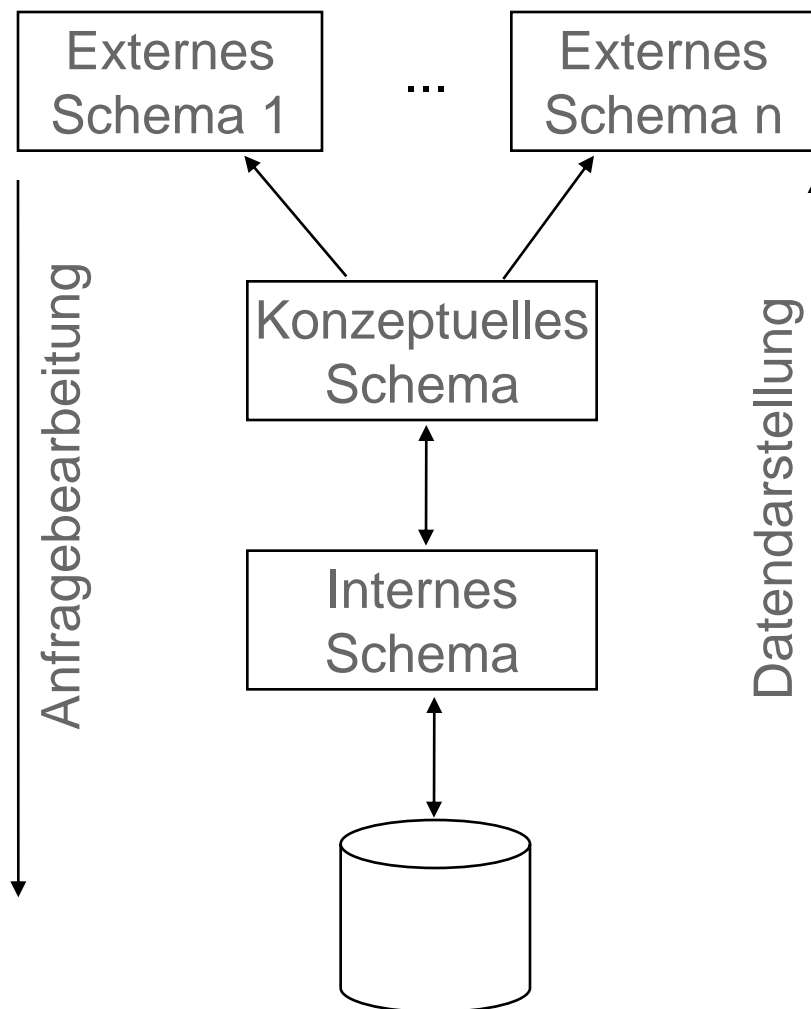
3-Ebenen-Architektur

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur



Ebenen-Architektur am Beispiel

- Konzeptuelle Sicht: Darstellung in Tabellen (Relationen)

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

Musiker	<u>MNr</u>	Name	Land
	103	Apocalyptica	Finnland
	104	Subway To Sally	Deutschland
	105	Rammstein	Deutschland

Album	<u>ANr</u>	Titel	Jahr	Genre	<u>MNr</u> → Musiker
	1014	Amplified	2006	Rock	103
	1015	Nord Nord Ost	2005	Rock	104
	1016	Rosenrot	2005	Rock	105
	1021	Engelskrieger	2003	Rock	104
	1025	Reflections	2006	Rock	103

Ebenen-Architektur am Beispiel

- Externe Sicht: Eine Relation mit Einschränkungen auf den sichtbaren Attributen

Grundlagen

Schema-
architektur

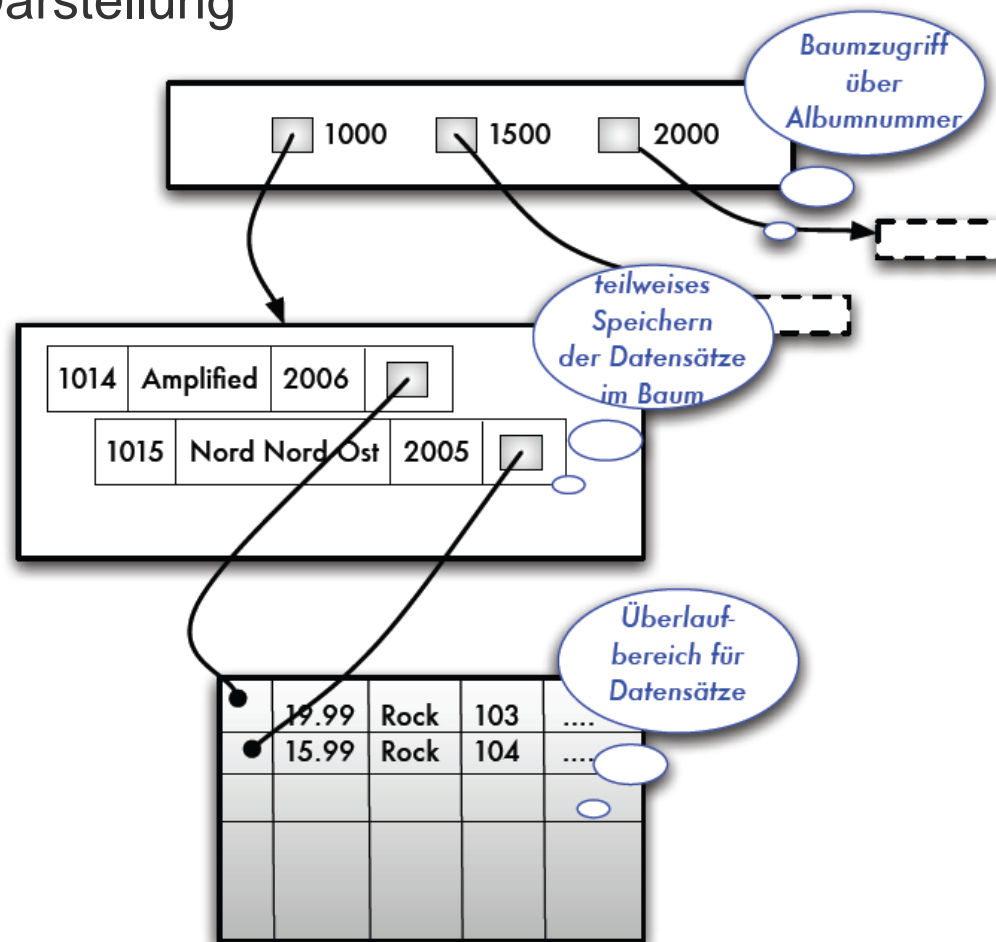
System-
architektur

Anwendungs-
architektur

<u>ANr</u>	Titel	Jahr	Genre	Musiker
1014	Amplified	2006	Rock	Apocalyptica
1015	Nord Nord Ost	2005	Rock	Subway To Sally
1016	Rosenrot	2005	Rock	Rammstein
1021	Engelskrieger	2003	Rock	Subway To Sally
1025	Reflections	2006	Rock	Apocalyptica

Ebenen-Architektur am Beispiel

- Interne Darstellung



Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

Schemaarchitektur (2)

- Trennung Schema – Instanz
 - Schema (Metadaten, Datenbeschreibungen)
 - Instanz (Anwenderdaten, Datenbankzustand)
- Datenbankschema besteht aus
 - Internem, konzeptuellen, externen Schema und den Anwendungsprogrammen
 - Für alle Schemata ist eine formale Beschreibung notwendig, um die Transformation von Anfragen bzw. Anfrageergebnissen zwischen den Schemata zu realisieren (Anfragebearbeitung/Datendarstellung)
- Im konzeptuellen Schema etwa
 - Strukturbeschreibungen
 - Integritätsbedingungen
 - Autorisierungsregeln

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

Datenunabhängigkeit (2)

2 Aspekte der Datenunabhängigkeit:

- **Physische Datenunabhängigkeit (Implementierungsunabhängigkeit)**
 - Änderungen der Dateiorganisation und Zugriffspfade haben keinen Einfluss auf das konzeptuelle Schema
- **Logische Datenunabhängigkeit (Anwendungsunabhängigkeit)**
 - Änderungen an externen Schemata (Sichten auf das konzeptuelle Schema) haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme (koppelt DB ab von Änderungen und Erweiterungen der Anwendungsschnittstelle)
 - (Auch:) Änderungen am konzeptuellen Schema und gewissen externen Schemata haben keine Auswirkungen auf andere externe Schemata und Anwendungsprogramme
 - Aber: Änderungen am konzeptuellen Schema nicht immer problemfrei (Beispiel für eine Änderung, von denen Anwendungsprogramme betroffen sind/ nicht betroffen sind?)

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur

System-Architekturen

- Systemarchitektur:
 - Beschreibung der Komponenten eines DBMS
 - Standardisierung der Schnittstellen zwischen Komponenten.
- Architekturvorschläge
 - ANSI-SPARC-Architektur
 - Drei-Ebenen-Architektur
 - Fünf-Schichten-Architektur
 - Beschreibt Transformationskomponenten im Detail
 - Vorlesung „Datenbankimplementierung und - Tuning“ (Sommersemester)

Grundlagen

Schema-
architektur

System-
architektur

Anwendungs-
architektur



ANSI-SPARC-Architektur

- ANSI: American National Standards Institute
- SPARC: Standards Planning and Requirement Committee
- Vorschlag von 1978
- Verfeinerung der Grobarchitektur
 - Interne Ebene verfeinert
 - Mehr Interaktive- und Programmierkomponenten
 - Schnittstellen bezeichnet und normiert

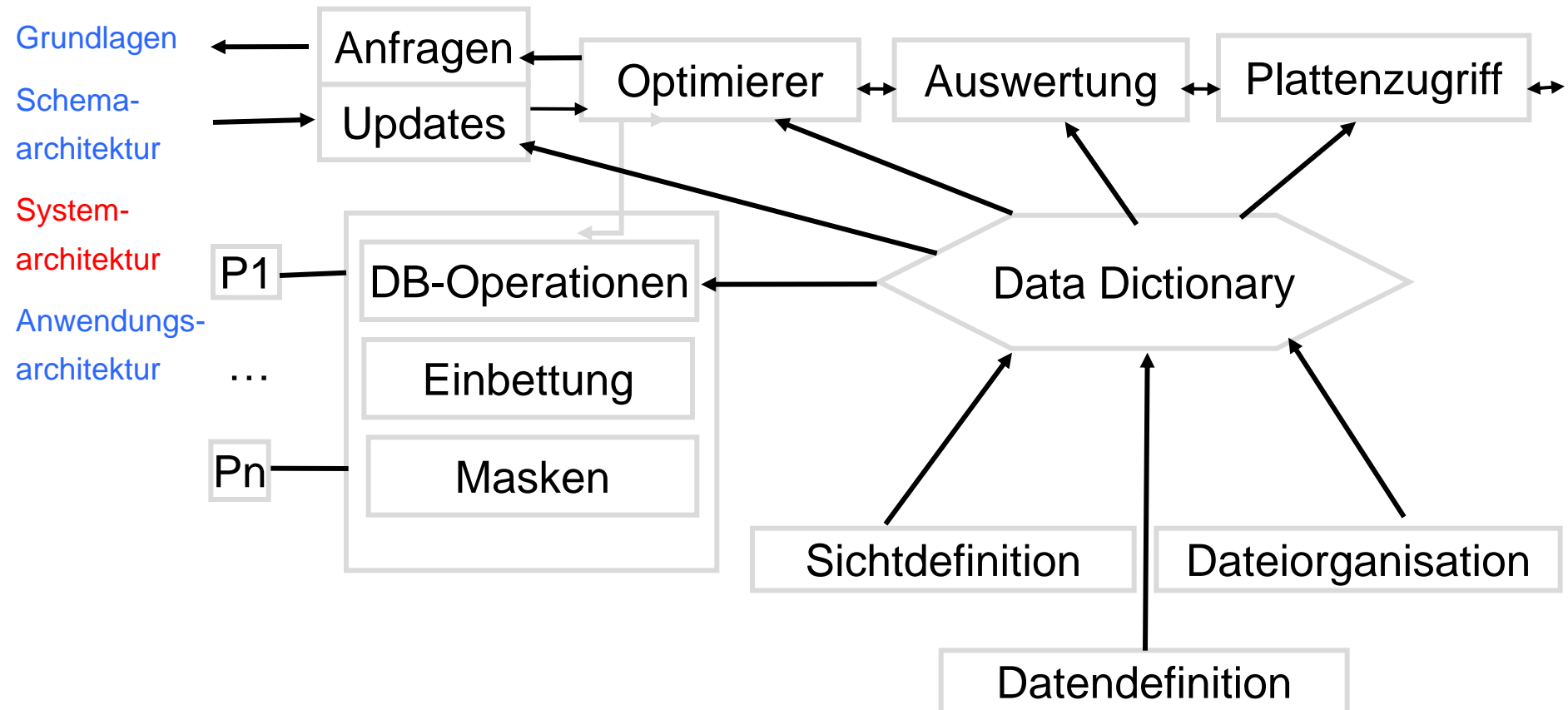
Grundlagen

Schema-
architektur

System-
architektur

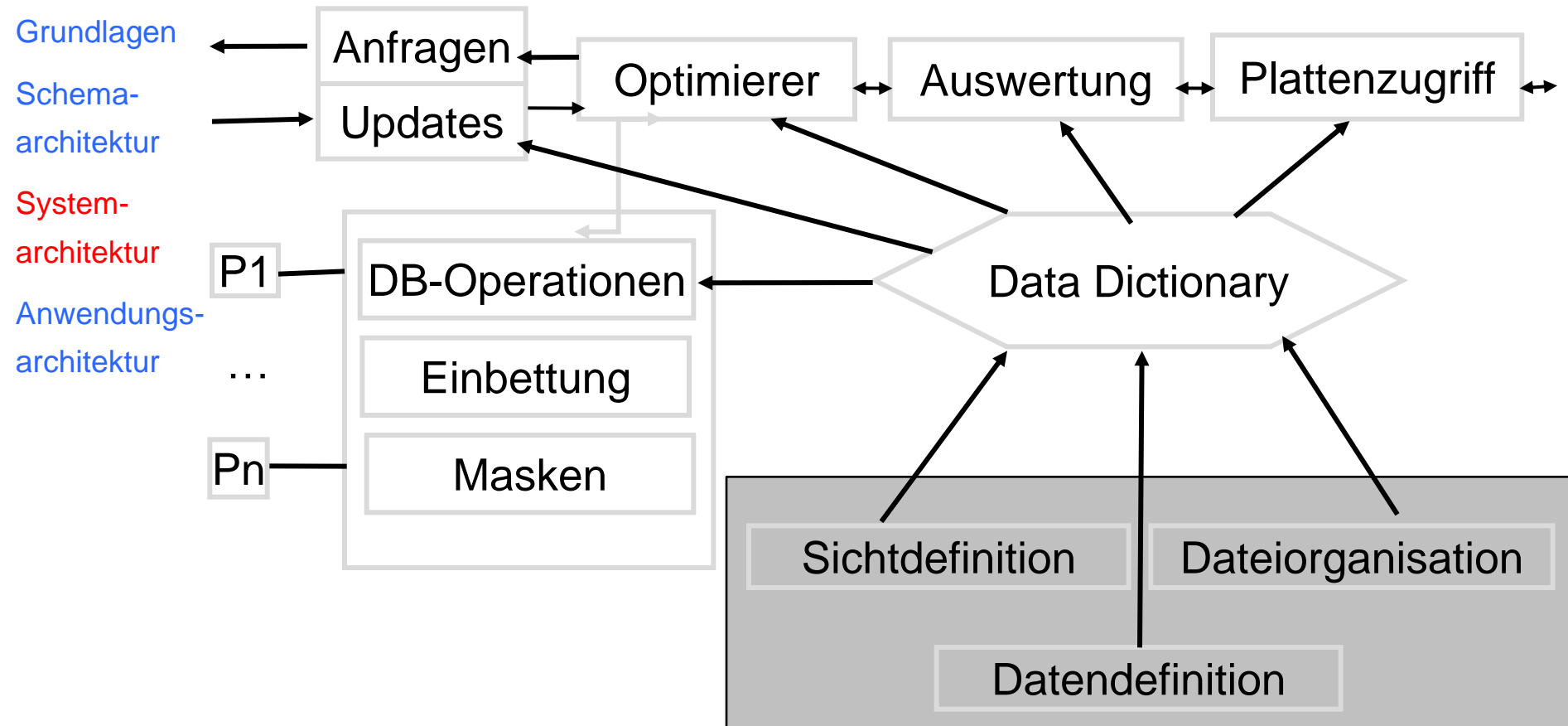
Anwendungs-
architektur

ANSI-SPARC-Architektur



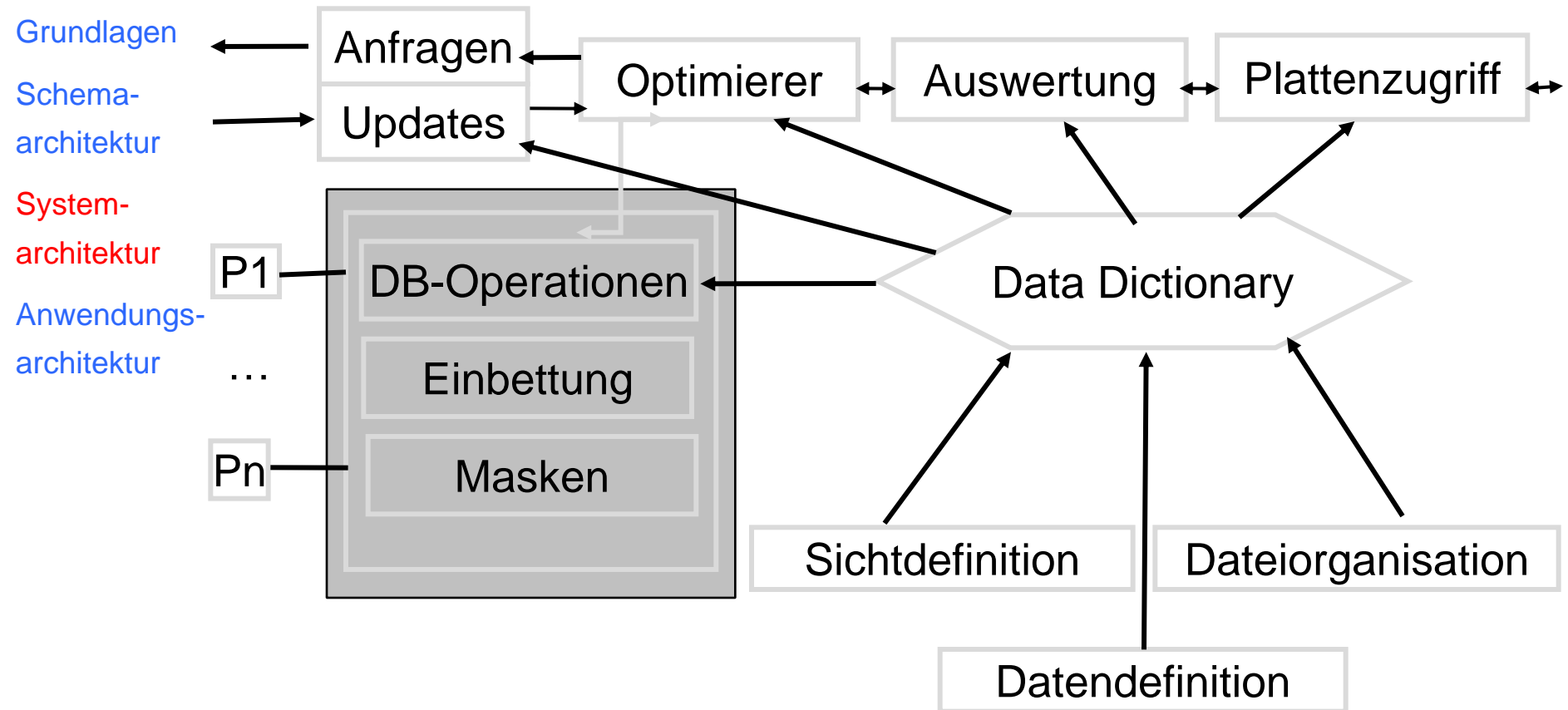
ANSI-SPARC-Architektur

- Definitionskomponenten: Datendefinition, Dateiorganisation, Sichtdefinition



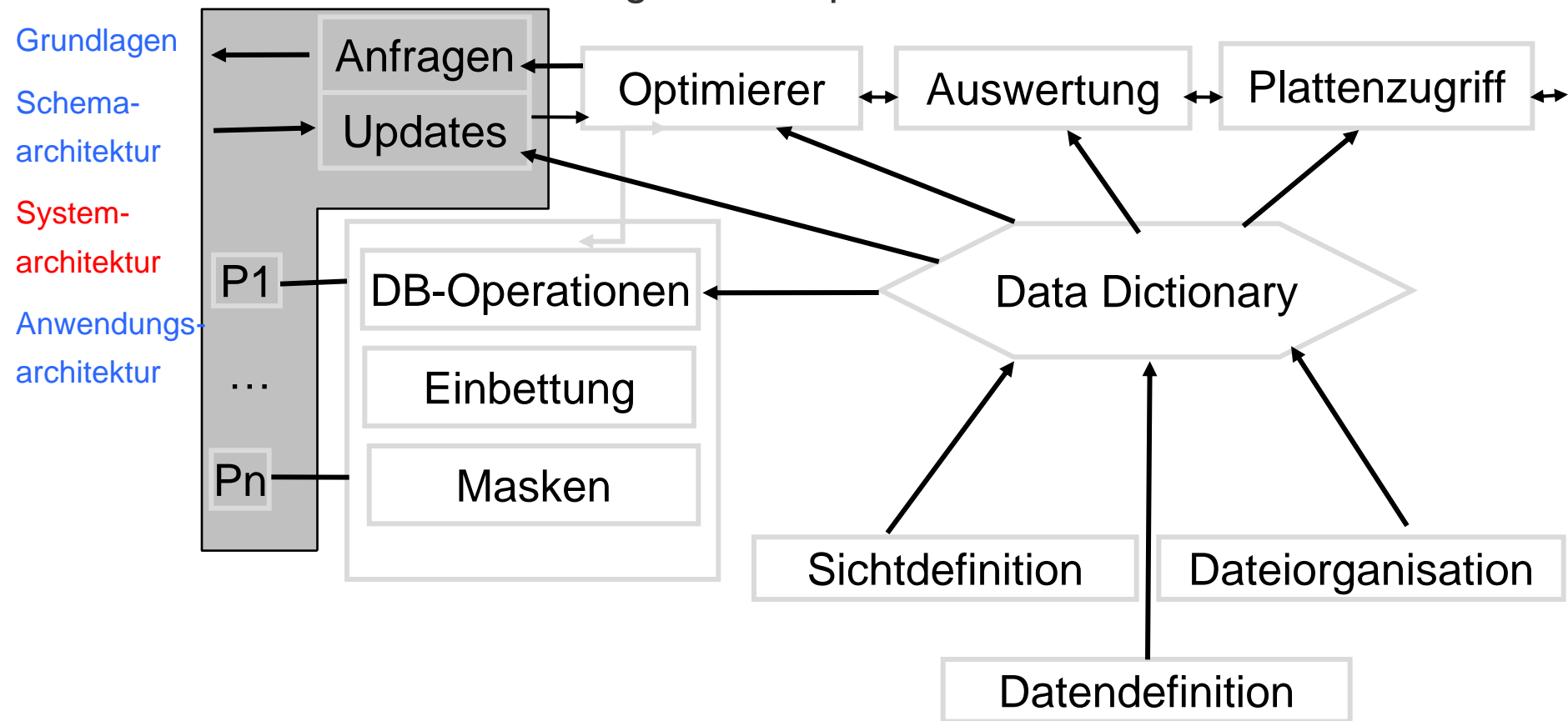
ANSI-SPARC-Architektur

- Programmierkomponenten: DB-Programmierung mit eingebetteten DB-Operationen



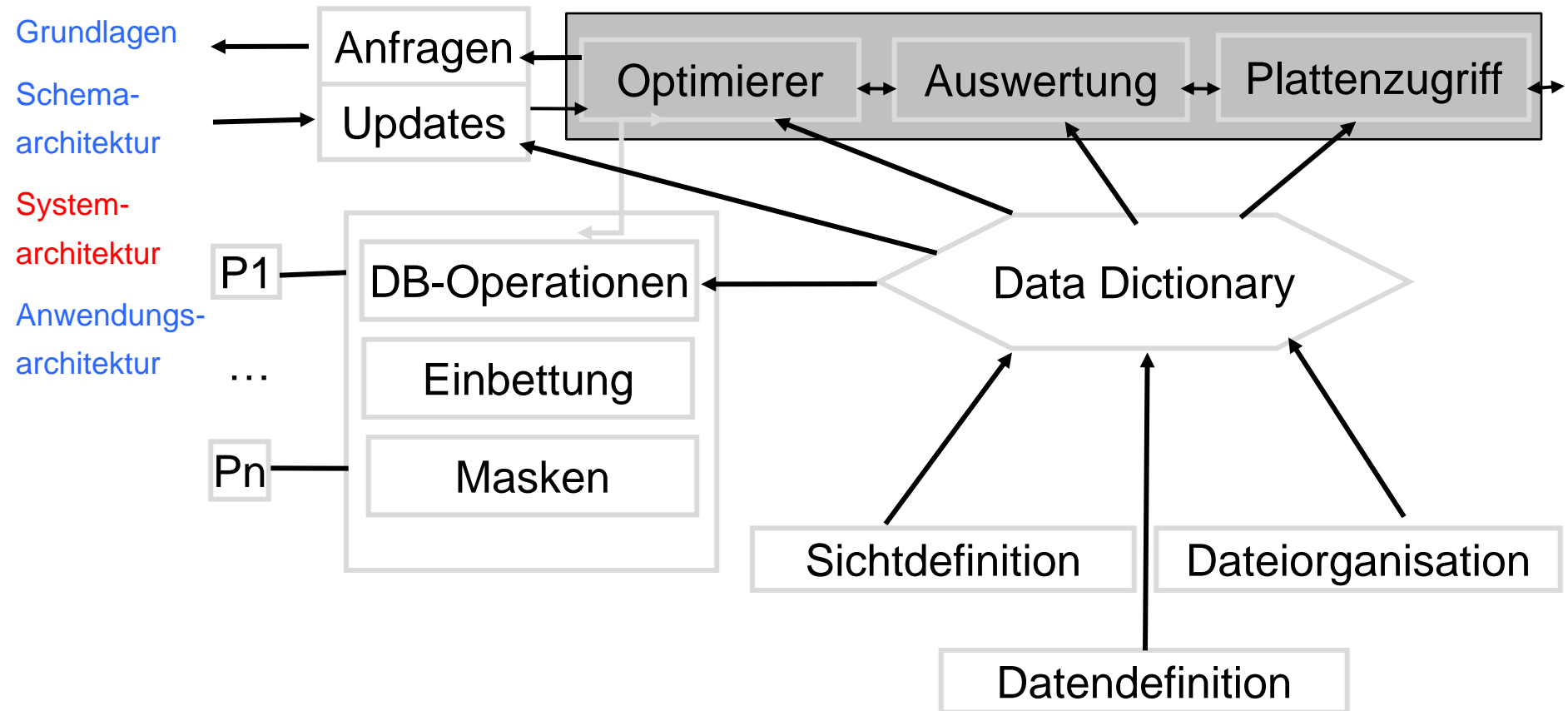
ANSI-SPARC-Architektur

- Benutzerkomponenten: Anwendungsprogramme, interaktive Anfragen und Updates



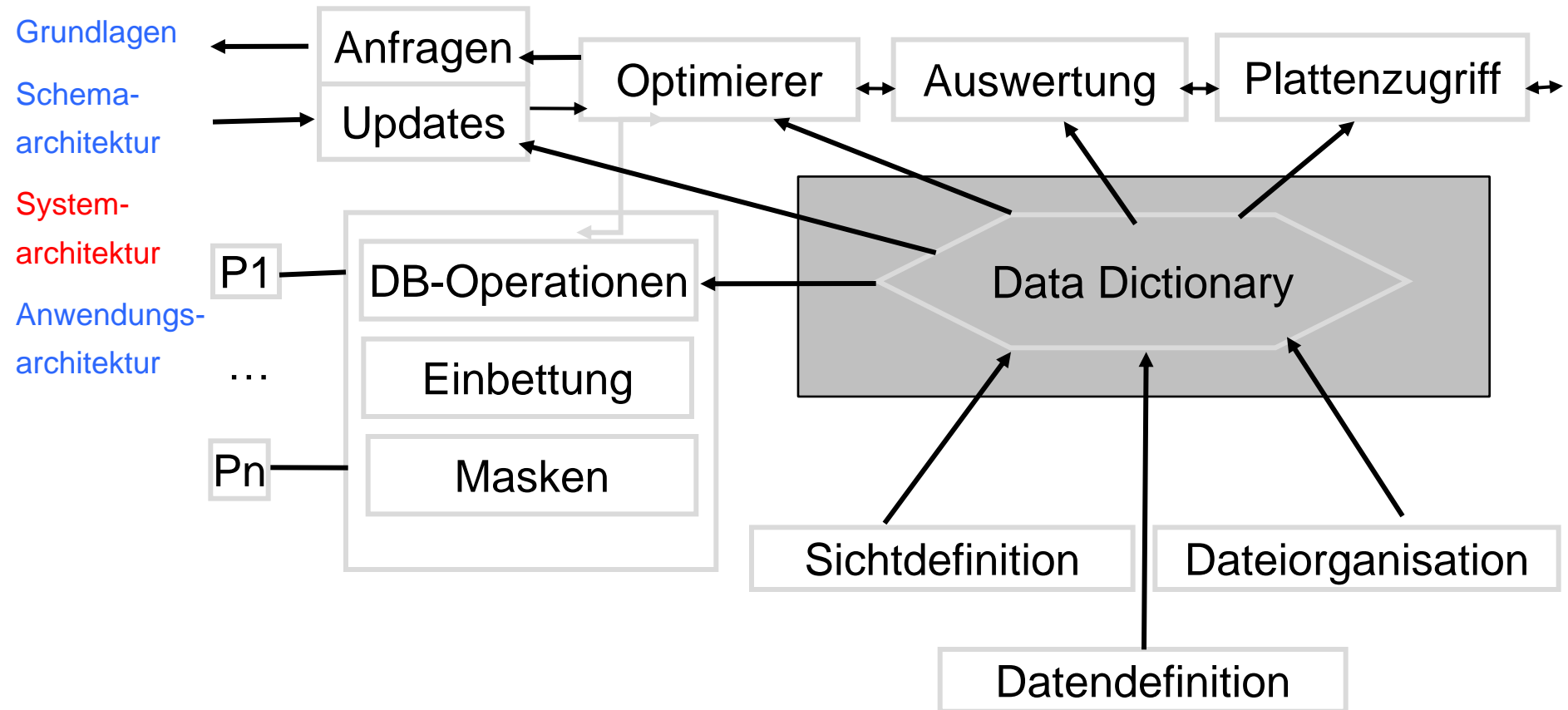
ANSI-SPARC-Architektur

- Transformationskomponenten: Optimierer, Auswertung, Plattenzugriffssteuerung



ANSI-SPARC-Architektur

- Data Dictionary: Verwaltung von Metadaten, Querschnittsfunktion

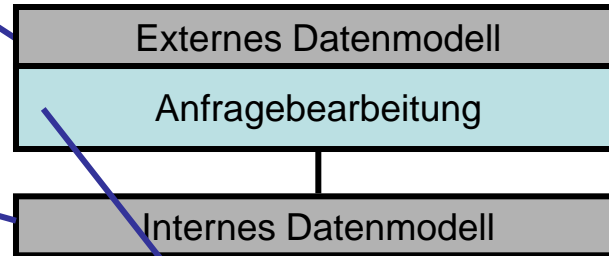


Fünf-Schichten-Architektur (1)

Datenmodell

Datentypen:
Satzmengen
Operatoren:
Operatoren auf Mengen

Datentypen:
Sätze und Satz­mengen
Operatoren:
Operatoren auf Sätzen



Aufgaben:

- Übersetzung von Anfragen auf Mengen in Folgen von Satz-Operationen
- Optimierung dieser Operationsfolgen
- Durchsetzung von Konsistenzbedingungen gemäß statischer Datenstruktur

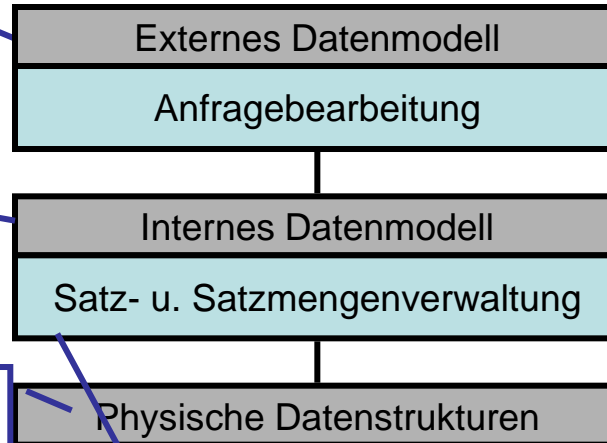
Fünf-Schichten-Architektur (2)

Datenmodell

Datentypen:
Satzmengen
Operatoren:
Operatoren auf Mengen

Datentypen:
Sätze und Satz­mengen
Operatoren:
Operatoren auf Sätzen

Datentypen:
Listen, Suchbäume, Hashtabellen, ...
Operatoren:
Listen: seq. Durchlauf
Bäume: seq. Durchlauf, gez. Suche
Hashtabellen: gezielte Suche



Aufgaben:

- Abbildung von Satz­mengen auf physische Datenstrukturen (Bäume, Hashtabellen, ...) gemäß **Zugriffsmuster**
- Abbildung von Satz- u. Satz­mengenoperatoren auf Operatoren der unterliegenden Datenstruktur

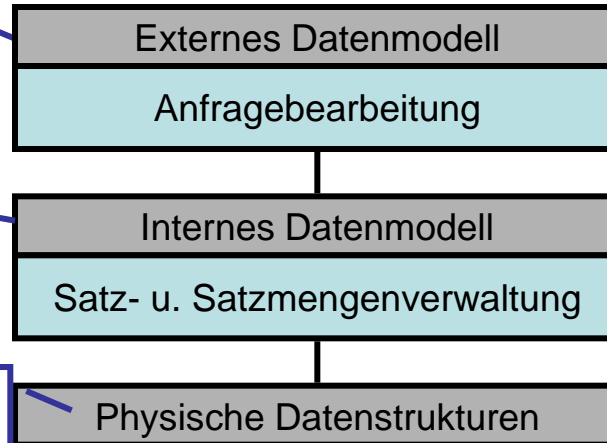
Fünf-Schichten-Architektur (2)

Datenmodell

Datentypen:
Satzmengen
Operatoren:
Operatoren auf Mengen

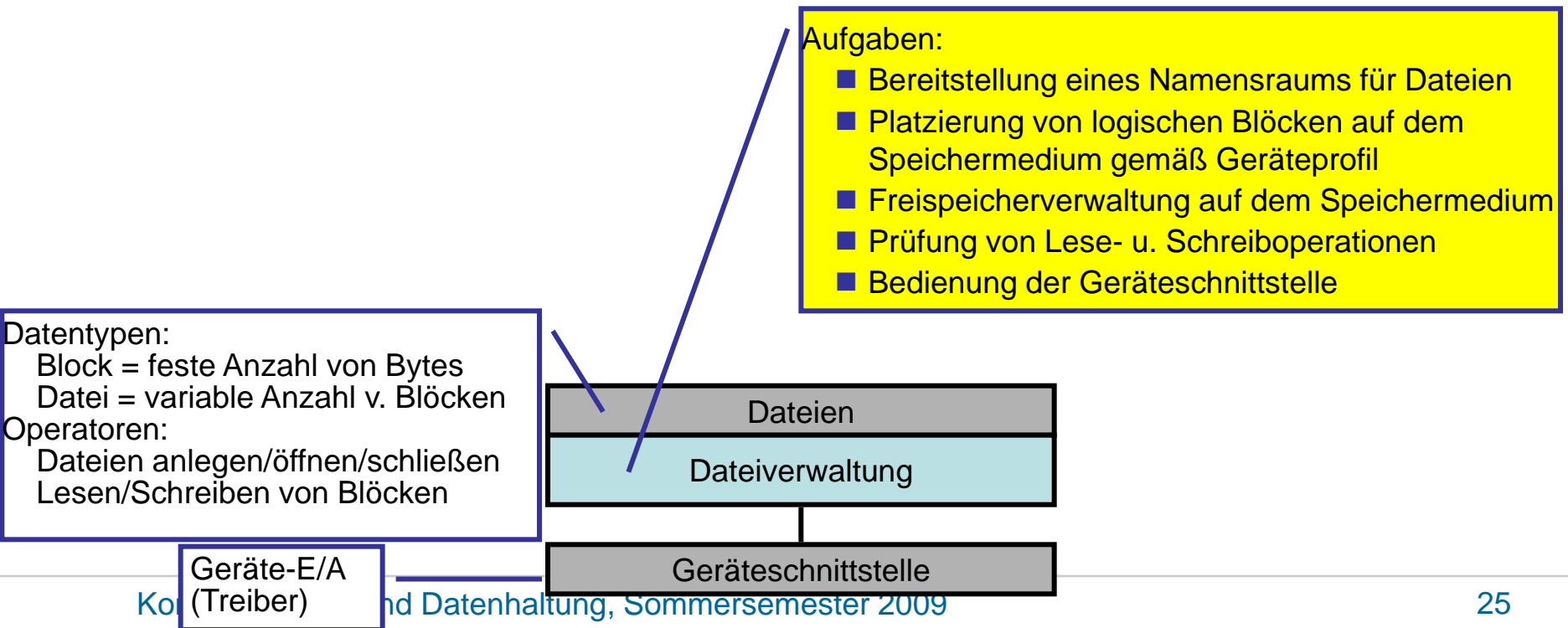
Datentypen:
Sätze und Satz­mengen
Operatoren:
Operatoren auf Sätzen

Datentypen:
Listen, Suchbäume, Hashtabellen, ...
Operatoren:
Listen: seq. Durchlauf
Bäume: seq. Durchlauf, gez. Suche
Hashtabellen: gezielte Suche



Fünf-Schichten-Architektur (3)

Datenmodell



Fünf-Schichten-Architektur (4)

Datenmodell

Aufgaben:

- Zuordnung von Segmenten zu Dateien
- Zuordnung von Seiten zu Dateiblöcken
- Anstoß und Überwachung von Seiten-Transfers zwischen Haupt- u. Hintergrundspeicher
- Minimierung von Transfer-Wartezeiten durch geeignete Pufferungs-Strategien gemäß **Zugriffsmuster**

Datentypen:

Seite = feste Anzahl von Bytes
Segment = var. Anzahl von Seiten

Operatoren:

Anforderung/Freigabe von Seiten
Segmente anlegen/öffnen/schließen

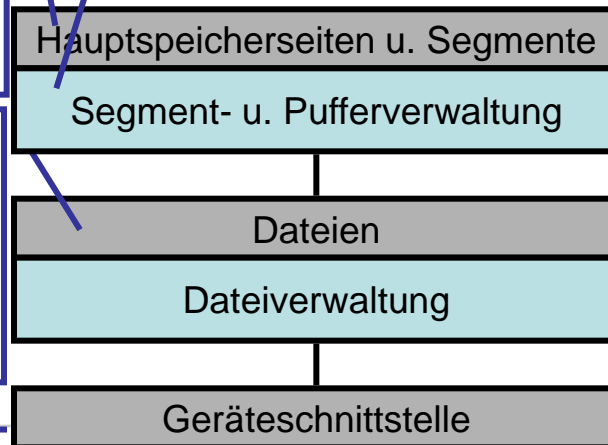
Datentypen:

Block = feste Anzahl von Bytes
Datei = variable Anzahl v. Blöcken

Operatoren:

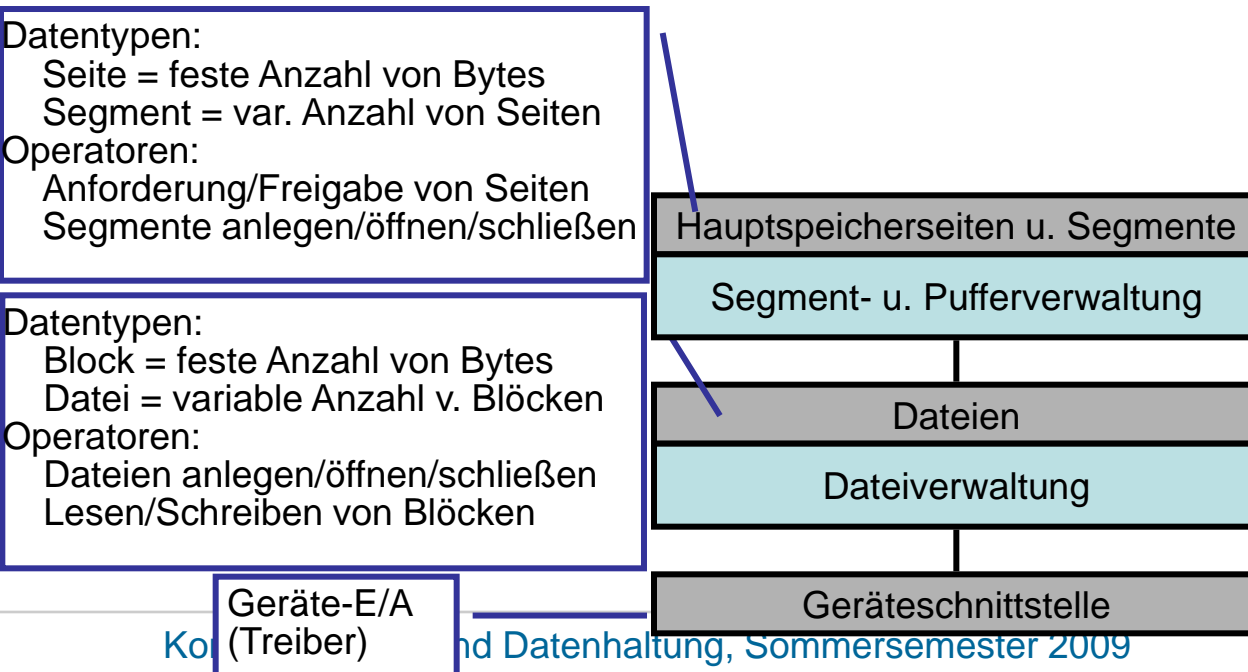
Dateien anlegen/öffnen/schließen
Lesen/Schreiben von Blöcken

Geräte-E/A
(Treiber)



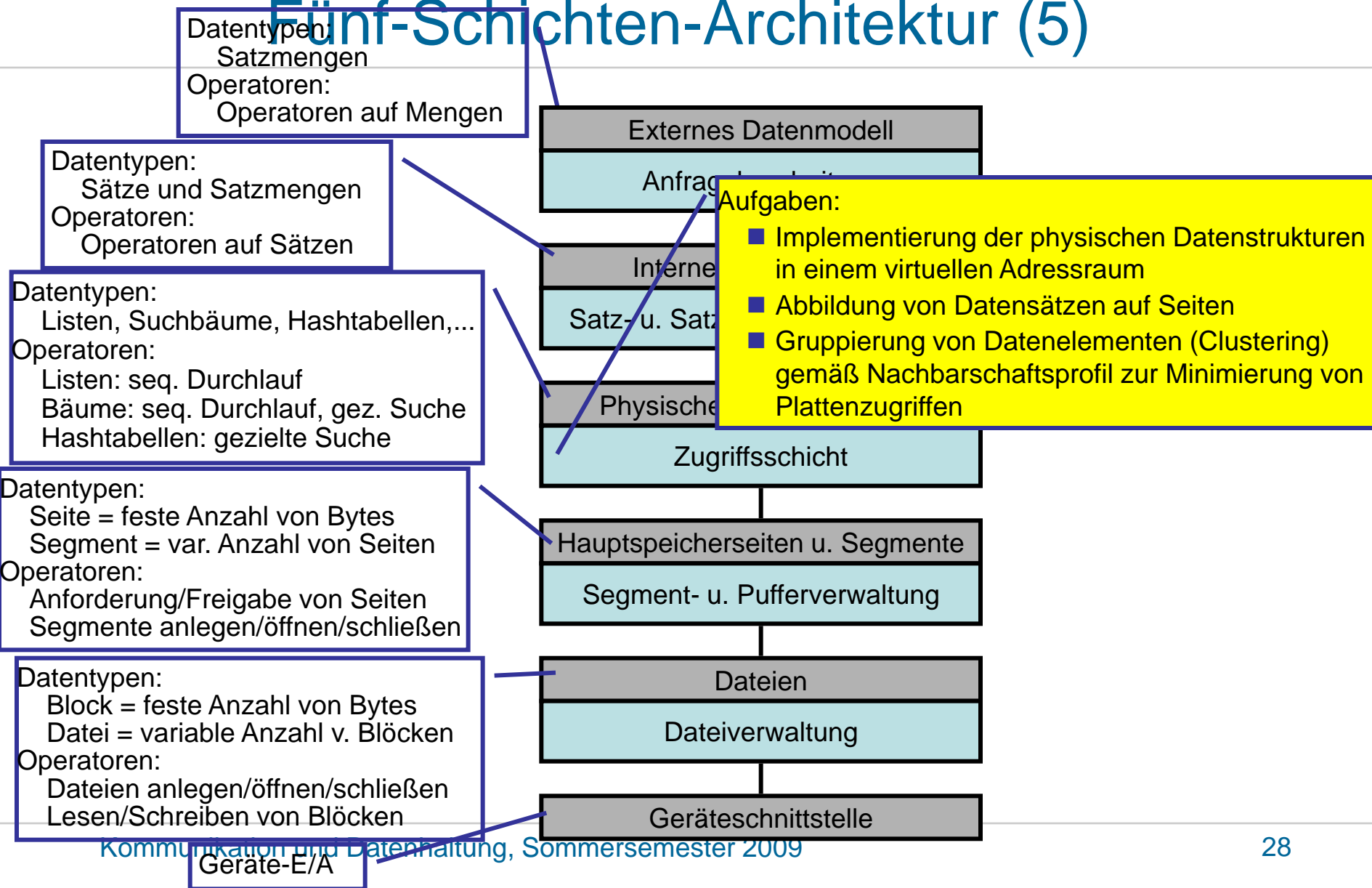
Fünf-Schichten-Architektur (4)

Datenmodell



Datenmodell

Fünf-Schichten-Architektur (5)



Datenmodell

Fünf-Schichten-Architektur (5)

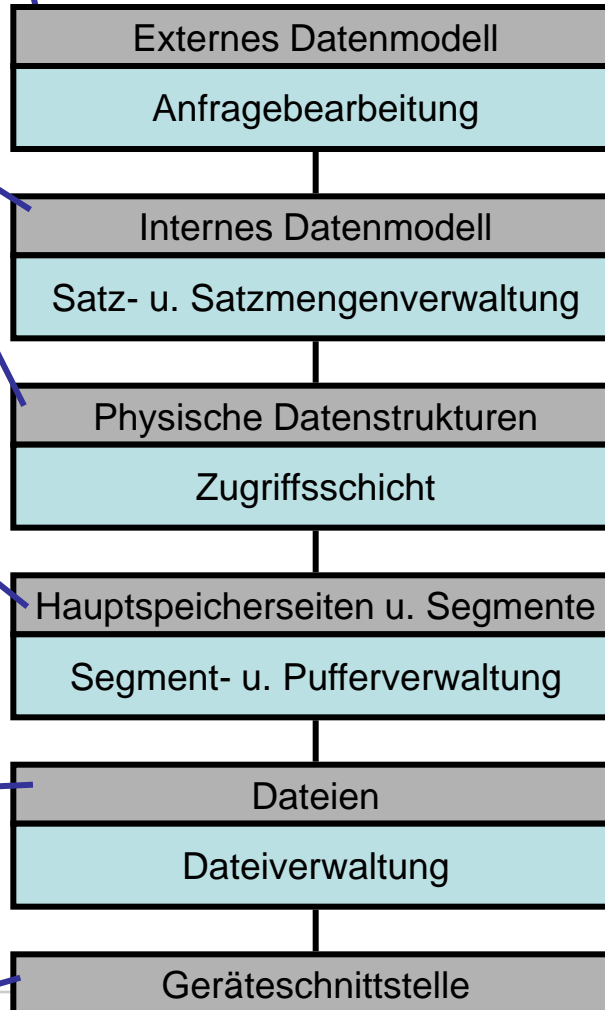
Datentypen:
Satzmengen
Operatoren:
Operatoren auf Mengen

Datentypen:
Sätze und Satz­mengen
Operatoren:
Operatoren auf Sätzen

Datentypen:
Listen, Suchbäume, Hashtabellen, ...
Operatoren:
Listen: seq. Durchlauf
Bäume: seq. Durchlauf, gez. Suche
Hashtabellen: gezielte Suche

Datentypen:
Seite = feste Anzahl von Bytes
Segment = var. Anzahl von Seiten
Operatoren:
Anforderung/Freigabe von Seiten
Segmente anlegen/öffnen/schließen

Datentypen:
Block = feste Anzahl von Bytes
Datei = variable Anzahl v. Blöcken
Operatoren:
Dateien anlegen/öffnen/schließen
Lesen/Schreiben von Blöcken

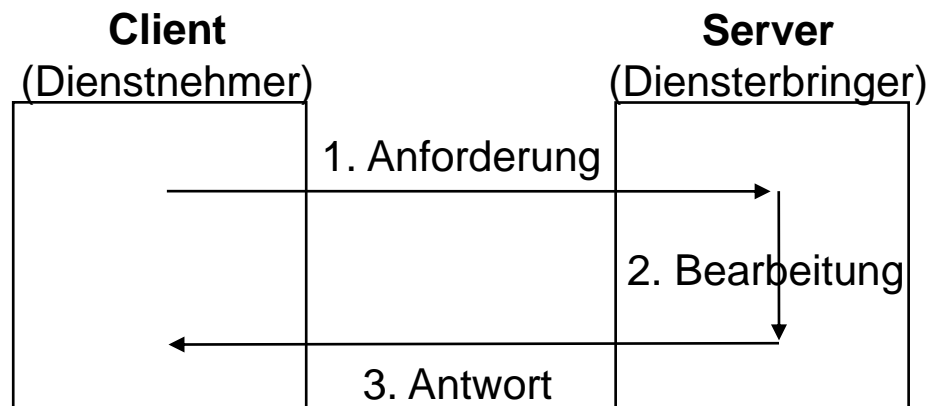


Diskussion (Fünf-Schichten-Architektur)

- Fehlende Komponenten:
 - Scheduler (TAV - Transaktionsverwaltung)
 - Logging/ Recovery-Verwalter
 - Archiv-Verwaltung (Backup)
 - Bisher: Nur Funktionalität angedeutet – offen:
 - Wie realisiert man die Abbildungen?
 - Vor allem: Performanz!
 - „Datenauftritt“ – Vorhaltung von in Kürze benötigten Daten
 - Vorhersage erforderlich (Kontextwissen ausnutzen; Zugriffsprofile)
- ⇒ Detaillierte Betrachtung in VL „Datenbankimplementierung und –Tuning“
- Um DBS auf Funktionalität und Leistung zu konfigurieren (Administrator) ist es unerlässlich, „Innere“ dieser Systeme zu kennen
 - Softwareengineering komplexer Systeme (Architektur, Maßnahmen zur Leistungssteigerung)

Anwendungsarchitekturen

- Architektur von Datenbank Anwendungen typischerweise auf Basis des Client-Server-Modells: Server = DBS



- Zentrale Kontrollinstanz vorteilhaft im Mehrbenutzerbetrieb
 - Vermeidet Redundanzen und Inkonsistenzen
 - Verringert Administrationsaufwand
- Clients entlasten Server
 - Aufbereitung der Daten, etc.

Anwendungsarchitekturen (2)

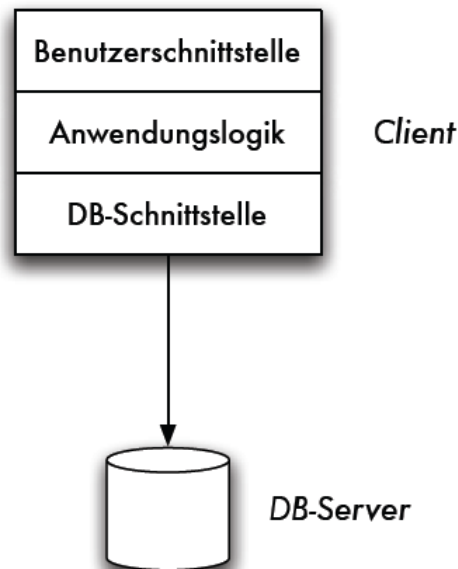
- Aufteilung der Funktionalität einer Anwendung
 - Präsentation und Benutzerinteraktion
 - Anwendungslogik („Business“-Logik)
 - Datenmanagementfunktionen (Anfragebearbeitung, TAV...)

Grundlagen

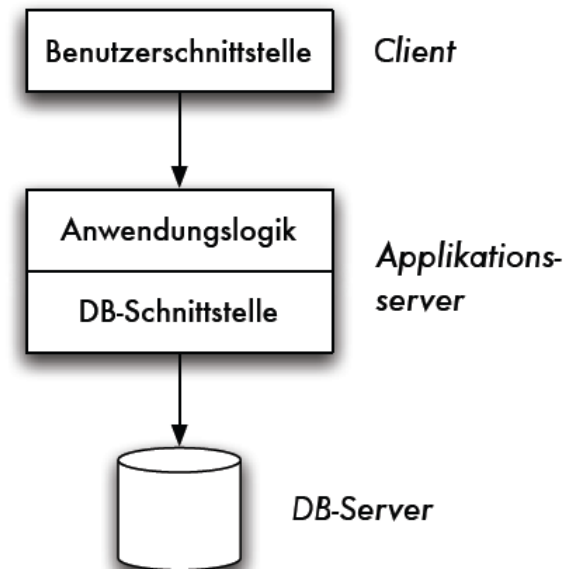
Schema-
architektur

System-
architektur

Anwendungs-
architektur



Zwei-Schichten-Architektur



Drei-Schichten-Architektur



Zusammenfassung

- Drei-Ebenen-Architektur
 - Aufteilung eines DB-Schemas: internes, konzeptuelles, externes
- Internes Schema
 - Beschreibung der systemspezifischen Realisierung einer DB
- Konzeptionelles Schema
 - Implementierungsunabhängige Modellierung
- Externes Schema
 - Modellierung der anwendungsspezifischen Sicht
- Datenunabhängigkeit
 - Entkopplung der Daten von Anwendungsprogrammen
- ANSI-SPARC-Architektur
 - Normungsvorschlag für Systemarchitektur von DBMS
- Fünf-Schichten-Architektur
 - Systembeschreibung: Transformationskomponenten
- Anwendungsarchitektur (Client-Server-Architektur)