MOBI-DBS-B: Datenbanksysteme Verteilte Datenbanken

Vorlesung Sommersemester 2019

Tanya Braun, Universität zu Lübeck

Lehrauftrag SoSe 19, Universität Bamberg

Erweiterung: Verteilte Datenbanken

Inhalte

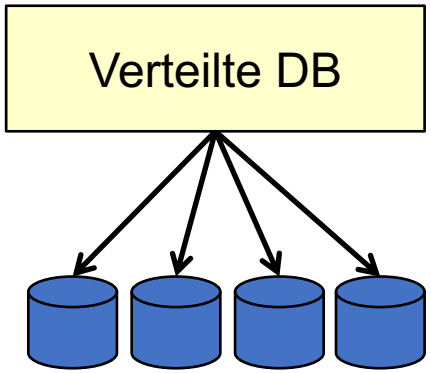
- Verteilung von Daten
  - Fragmentierung
    - Horizontal
    - Vertikal
    - Abgeleitet
  - Replikation
  - Allokation
  - … daraus folgend
    - Transparenz
    - Herausforderungen der Transaktionskontrolle
- Integration
  - Föderierte DBS
  - Migration

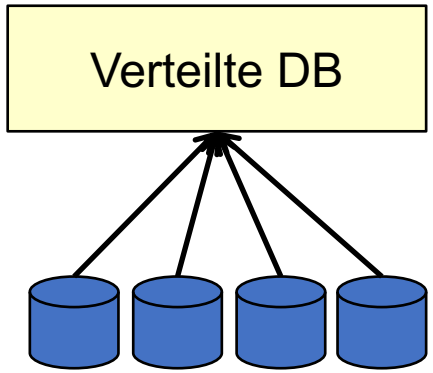
Kompetenzen

- Probleme und Möglichkeiten in verteilten Datenbanken erkennen

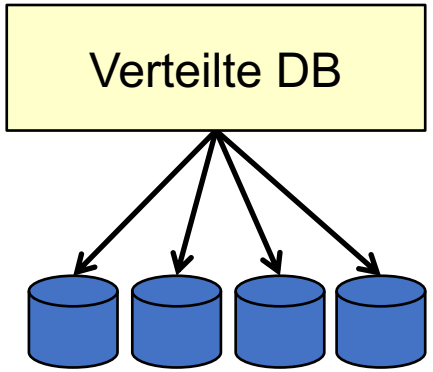
Motivation für verteilte Datenbanken

- Verteilung/Dezentralisierung
  - Vor allem bei Neuentwicklungen
  - (Verteilte) Realisierung von Anwendungssystemen auf der Basis verteilter (DB-)Systeme
  - (Kontrolliert) verteilte Datenspeicherung zur Lastverteilung, Nutzung von Parallelität zur Verkürzung von Antwortzeiten, Erhöhung der Ausfallsicherheit
- Integration
  - Bei bestehenden Systemen
  - Verteilt gespeicherte und unabhängig voneinander verwaltete, aber inhaltlich zusammengehörige Daten logisch zusammenbringen
  - Einheitlichen Zugriff auf Gesamtdatenbestand ermöglichen





Fragmentierung, Replikation, Allokation



Verteilung und Dezentralisierung

Aufbau eines verteilten DBMS (VDBMS)

- Globales Schema
  - Relationales Schema wie zuvor
- Fragmentierungsschema
  - Verteilung der Daten in Fragmente
- Zuordnungsschema
  - Physische Zuordnung (Allokation) der Fragmente auf lokale DBs
  - Möglicherweise mit Replikation
- Lokales Schema/DBMS/DB
  - DB: Menge der lokal gespeicherten Daten
  - DBMS: Lokales System zur Verwaltung und Anfragebeantwortung
  - Schema: Lokale Sicht auf die Daten in lokaler DB

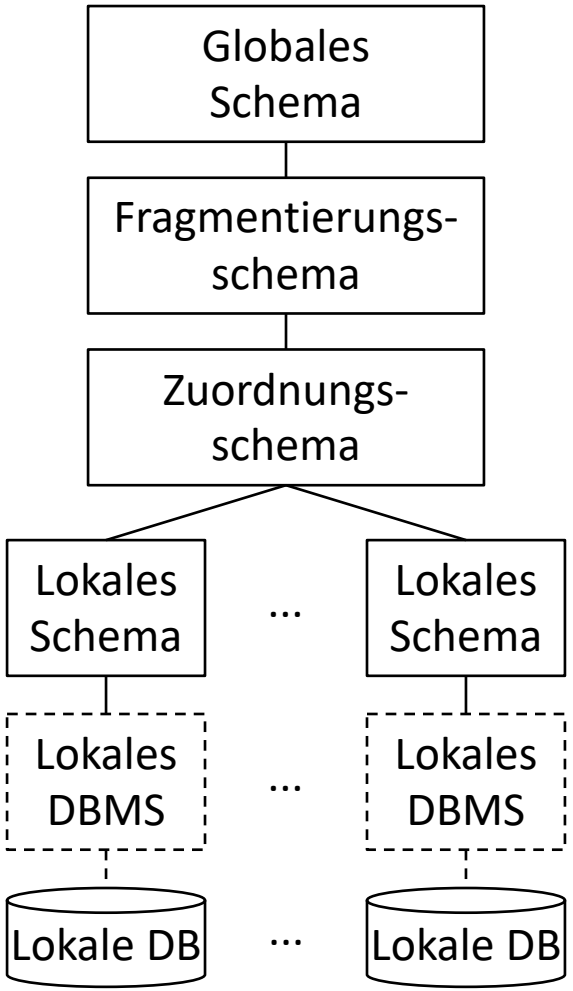
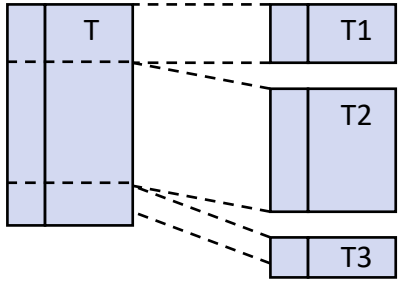


Abbildung nach Kemper & Eickler

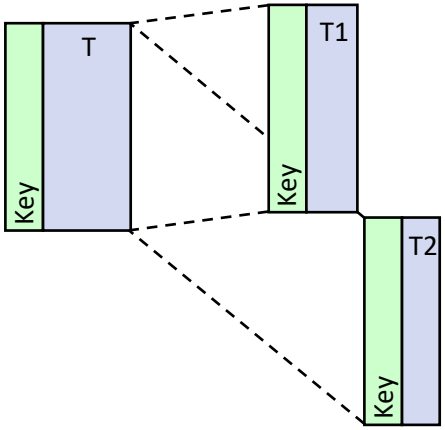
Fragmentierung

- Ziel: Verteilung von Daten
- Arten der Fragmentierung
  - Horizontal – entlang der Tabellenzeilen
  - Vertikal – entlang der Tabellenspalten
  - Gemischt – Verschachtelung von Fragmentierung
  - Abgeleitet – anhand von Bedingungen



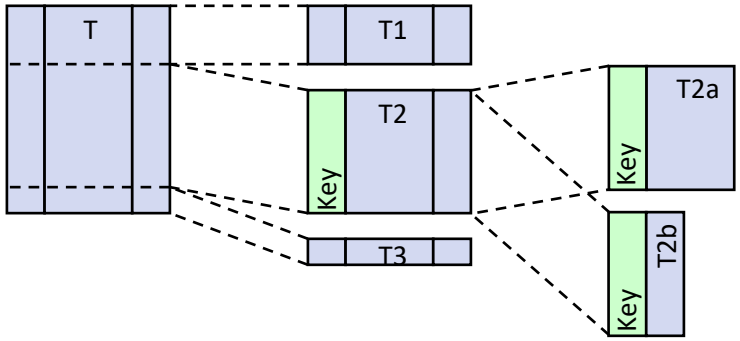
Horizontale Fragmentierung

Aufteilung durch Selektion



Vertikale Fragmentierung

Aufteilung durch Projektion



Beispiel für gemischte Fragmentierung

Aufteilung durch verschachtelte Selektion und Projektion

Beispiel für horizontale Fragmentierung

Profs

PersNr	Name	Rang	Raum	Fakultaet	Gehalt	St.Klasse
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2127	Kopernikus	C3	310	Physik	65000	5
2133	Popper	C3	52	Philosophie	68000	1
2134	Augustinus	C3	309	Theologie	55000	5
2136	Curie	C4	36	Physik	95000	3
2137	Kant	C4	7	Philosophie	98000	1

Profs =
PhilProfs
∪ PhysProfs
∪ TheoProfs

PhilProfs =
$\sigma_{Fakultaet='Philosophie'}(Profs)$

PhilProfs

PersNr	Name	Rang	Raum	Fakultaet	Gehalt	St.Klasse
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2133	Popper	C3	52	Philosophie	68000	1
2137	Kant	C4	7	Philosophie	98000	1

PhysProfs =
$\sigma_{Fakultaet='Physik'}(Profs)$

PhysProfs

PersNr	Name	Rang	Raum	Fakultaet	Gehalt	St.Klasse
2127	Kopernikus	C3	310	Physik	65000	5
2136	Curie	C4	36	Physik	95000	3

TheoProfs =
$\sigma_{Fakultaet='Theologie'}(Profs)$

TheoProfs

PersNr	Name	Rang	Raum	Fakultaet	Gehalt	St.Klasse
2134	Augustinus	C3	309	Theologie	55000	5

Beispiel für vertikale Fragmentierung

Profs

PersNr	Name	Rang	Raum	Fakultaet	Gehalt	St.Klasse
2125	Sokrates	C4	226	Philosophie	85000	1
2126	Russel	C4	232	Philosophie	80000	3
2127	Kopernikus	C3	310	Physik	65000	5
2133	Popper	C3	52	Philosophie	68000	1
2134	Augustinus	C3	309	Theologie	55000	5
2136	Curie	C4	36	Physik	95000	3
2137	Kant	C4	7	Philosophie	98000	1

Profs = ProfVerw ⋈ Pfs

ProfVerw = $\pi_{PersNr,Name,Gehalt,Steuerklasse}(Profs)$

Pfs = $\pi_{PersNr,Name,Rang,Raum,Fakultaet}(Profs)$

ProfVerw

PersNr	Name	Gehalt	St.Klasse
2125	Sokrates	85000	1
2126	Russel	80000	3
2127	Kopernikus	65000	5
2133	Popper	68000	1
2134	Augustinus	55000	5
2136	Curie	95000	3
2137	Kant	98000	1

Pfs

PersNr	Name	Rang	Raum	Fakultaet
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2127	Kopernikus	C3	310	Physik
2133	Popper	C3	52	Philosophie
2134	Augustinus	C3	309	Theologie
2136	Curie	C4	36	Physik
2137	Kant	C4	7	Philosophie

Beispiel für gemischte Fragmentierung

Pfs

PersNr	Name	Rang	Raum	Fakultaet
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2127	Kopernikus	C3	310	Physik
2133	Popper	C3	52	Philosophie
2134	Augustinus	C3	309	Theologie
2136	Curie	C4	36	Physik
2137	Kant	C4	7	Philosophie

$ProfVerw = \pi_{PersNr,Name,Gehalt,Steuerklasse}(Profs)$

$Pfs = \pi_{PersNr,Name,Rang,Raum,Fakultaet}(Profs)$

$Profs = ProfVerw \bowtie (PhilPfs \cup PhysPfs \cup TheoPfs)$

$PhilPfs = \sigma_{Fakultaet='Philosophie'}(Pfs)$

PhilPfs

PersNr	Name	Rang	Raum	Fakultaet
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2133	Popper	C3	52	Philosophie
2137	Kant	C4	7	Philosophie

$PhysPfs = \sigma_{Fakultaet='Physik'}(Pfs)$

PhysPfs

PersNr	Name	Rang	Raum	Fakultaet
2127	Kopernikus	C3	310	Physik
2136	Curie	C4	36	Physik

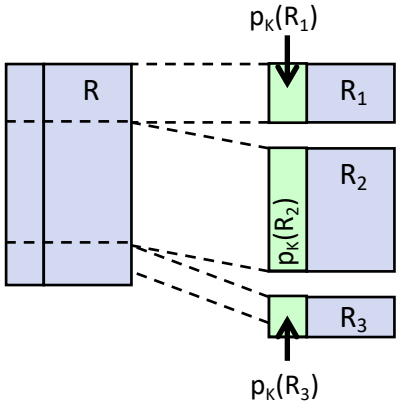
$TheoPfs = \sigma_{Fakultaet='Theologie'}(Pfs)$

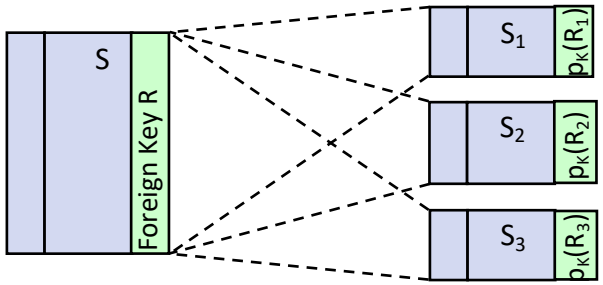
TheoPfs

PersNr	Name	Rang	Raum	Fakultaet
2134	Augustinus	C3	309	Theologie

Abgeleitete Fragmentierung

- Relation S wird auf Basis der Fragmentierung der Relation R (abgeleitet) fragmentiert
  - Fremdschlüssel B in S auf Primärschlüssel A in R
- R durch Selektion horizontal in Fragmente R_i zerlegt
  - Referenzpartitionierung für S
- S aufgrund Fremdschlüsselbeziehungen partitioniert
  - Primärschlüssel der Fragmente $p_K(R_i)$
  - $S_i = S \ltimes_{S.B=pK(Ri)} p_K(R_i)$
- $p_K(R_i)$ bilden disjunkte Mengen → auch S_i disjunkt





Beispiel für abgeleitete Fragmentierung

- Geg. horizontale Fragmentierung der Relation Pfs
  - Entspricht der Relation R
  - Disjunkte Mengen p_K(PersNr) der Primärschlüssel von PhilPfs, PhilPfs und PhysPfs

PhilPfs

PersNr	Name	Rang	Raum	Fakultaet
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2133	Popper	C3	52	Philosophie
2137	Kant	C4	7	Philosophie

p_K(PhilPfs) →

PersNr
2125
2126
2133
2137

PhysPfs

PersNr	Name	Rang	Raum	Fakultaet
2127	Kopernikus	C3	310	Physik
2136	Curie	C4	36	Physik

p_K(PhysPfs) →

PersNr
2127
2136

TheoPfs

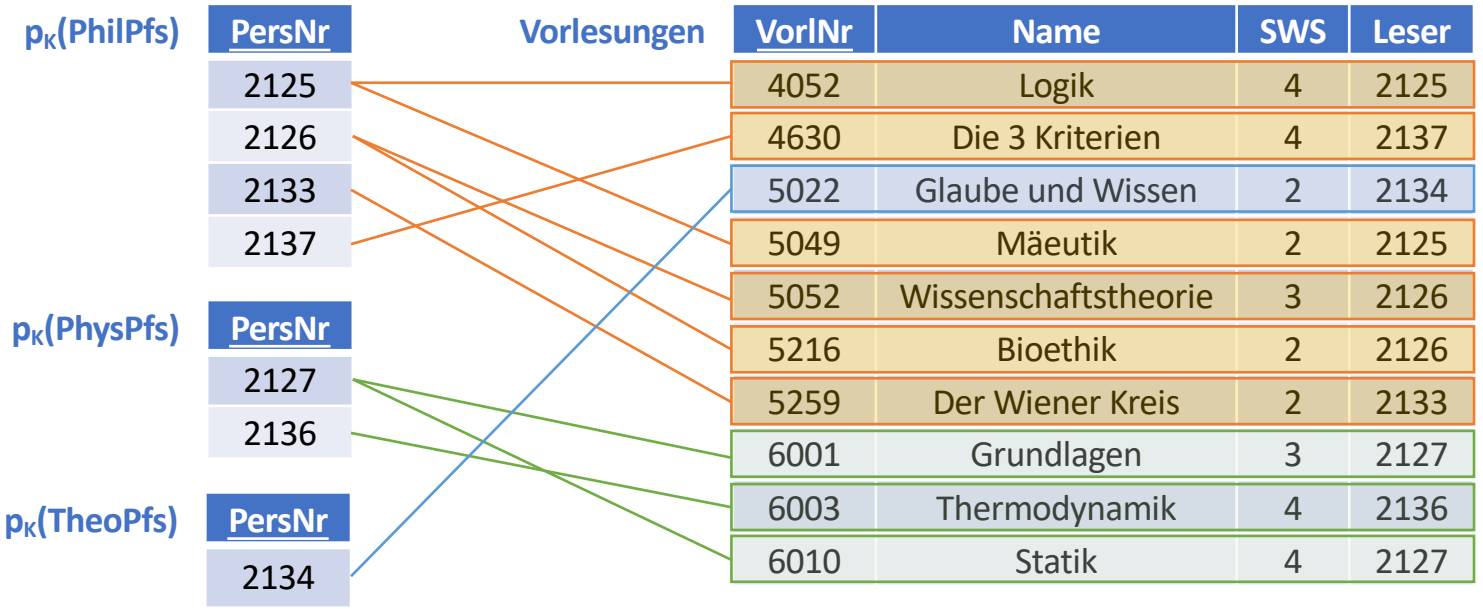
PersNr	Name	Rang	Raum	Fakultaet
2134	Augustinus	C3	309	Theologie

p_K(TheoPfs) →

PersNr
2134

Beispiel für abgeleitete Fragmentierung

- Geg. horizontale Fragmentierung der Relation Pfs
- Ableitung der Fragmentierung der Relation Vorlesungen
  - Vorlesungen entspricht S, Leser Fremdschlüssel auf PersNr



Beispiele Fragmentierungen

ProfVerw

PersNr	Name	Gehalt	St.klasse
2125	Sokrates	85000	1
2126	Russel	80000	3
2127	Kopernikus	65000	5
2133	Popper	68000	1
2134	Augustinus	55000	5
2136	Curie	95000	3
2137	Kant	98000	1

Profs = ProfVerw ⋈ (PhilPfs ∪ PhysPfs ∪ TheoPfs)
Vorlesungen = PhilVorl ∪ PhysVorl ∪ TheoVorl

PhilPfs

PersNr	Name	Rang	Raum	Fakultaet
2125	Sokrates	C4	226	Philosophie
2126	Russel	C4	232	Philosophie
2133	Popper	C3	52	Philosophie
2137	Kant	C4	7	Philosophie

PhysPfs

PersNr	Name	Rang	Raum	Fakultaet
2127	Kopernikus	C3	310	Physik
2136	Curie	C4	36	Physik

TheoPfs

PersNr	Name	Rang	Raum	Fakultaet
2134	Augustinus	C3	309	Theologie

PhilVorl

VorlNr	Name	SWS	Leser
4052	Logik	4	2125
4630	Die 3 Kriterien	4	2137
5049	Mäeutik	2	2125
5052	Wissenschaftstheorie	3	2126
5216	Bioethik	2	2126
5259	Der Wiener Kreis	2	2133

PhysVorl

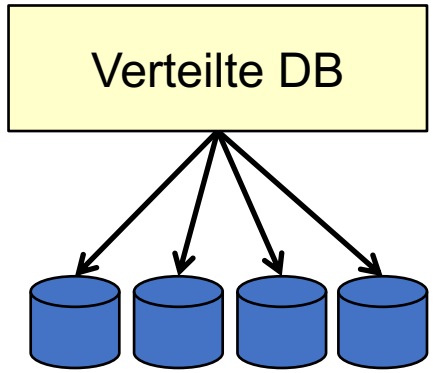
VorlNr	Name	SWS	Leser
6001	Grundlagen	3	2127
6003	Thermodynamik	4	2136
6010	Statik	4	2127

TheoVorl

VorlNr	Name	SWS	Leser
5022	Glaube und Wissen	2	2134

Fragmentierung

- Motivation
  - Lastverteilung
  - Nutzung von Parallelität zur Verkürzung von Antwortzeiten
- Art der Fragmentierung abhängig von häufigen Anfragen und Zugriffsmustern



- Korrektheit der Fragmentierung
  - Original-Relation lässt sich aus den Fragmenten wiederherstellen
  - Jedes Tupel ist einem Fragment zugeordnet
  - Die Fragmente überlappen sich nicht
- Herausforderungen
  - DBS muss Verteilung der Daten kennen und bei Anfragen berücksichtigen
    - An welche DB muss welche Teilanfrage?
    - Wie müssen Ergebnismengen zusammengefügt werden?
    - Was passiert, wenn eine DB ausfällt?

Replikation

- (Logisches) Duplizieren von Daten nach strategischen Gesichtspunkten

- Motivation: Erhöhung der …
  - Effizienz (schnellerer Zugriff „vor Ort statt entfernt“)
  - Ausfallsicherheit und Verfügbarkeit (Replikat als „Sicherungskopie“)
  - Autonomie (Unabhängigkeit von ansonsten nur einmal verfügbaren Daten)

- Zielkonflikte bei der Replikation
  - Deutliche Erhöhung der Zugriffseffizienz, Verfügbarkeit und Autonomie
    $\Rightarrow$ Große Anzahl von Replikaten auf möglichst vielen Knoten
  - Erhaltung der Datenkonsistenz
    $\Rightarrow$ Alle Replikate möglichst synchron aktualisieren
  - Erhöhung der Effizienz bei Änderungsoperationen
    $\Rightarrow$ Wenige Replikate wünschenswert; schneller synchron aktualisierbar

Allokation

- (Physische) gezielte Zuordnung von Daten auf Rechner
- Die (physisch orientierte) Allokation legt fest …
  - auf welchem Rechner ein Fragment gehalten wird
  - welche Fragmente auf welchen Rechnern repliziert gespeichert werden
- Physische Umsetzung der Überlegung aus Fragmentierung und Replikation
- Bei der Allokation zu berücksichtigende Aspekte …
  - Effizienz
    - Minimierung von Zugriffskosten für Remote-Zugriffe
    - Vermeidung von „Flaschenhälsen“
      (Übertragungskapazität im Netz, Leistung einzelner Rechenknoten, ...)
  - Datensicherheit
    - Auswahl von Knoten hinsichtlich Verlässlichkeit
    - Redundante Speicherung von Daten

Fortsetzung Beispiel

- Allokation ohne Replikation

Lokale DB/Station	Bemerkung	Zugeordnete Fragmente
DB_{Verw}	Verwaltungsrechner	{ProfVerw}
DB_{Philo}	Dekanat Physik	{PhilVorl, PhilPrfs}
DB_{Phys}	Dekanat Philosophie	{PhysVorl, PhysPrfs}
DB_{Theo}	Dekanat Theologie	{TheoVorl, TheoPrfs}

- Allokation mit Replikation, Beispiele:
  - Erhöhung der Ausfallsicherheit:
    Zweite DB_{Verw}-Station
  - Prüfungsamt benötigt Vorlesungsinformation:
    Weitere Station DB_{Vorl} mit den wiedervereinigten Vorlesungsfragmenten

Transparenz in verteilten Datenbanken

Was muss der Nutzer über die verteilte DB wissen?

- Fragmentierungstransparenz
  - Nutzer stellen Anfragen an das globale Schema, haben kein Wissen über Fragmentierung/Allokation
  - VDBMS muss Anfragen und Änderungen korrekt auf Fragmenten und in Stationen umsetzen
- Allokationstransparenz
  - Nutzer müssen die Fragmentierung kennen, aber nicht die Allokation
  - Anfragen an Fragmente/lokale Schemata
- Lokale Schema-Transparenz
  - Nutzer müssen Station kennen, auf dem Fragment liegt
  - Transparenz bezieht sich darauf, dass zumindest alle Stationen dasselbe Datenmodell und dieselbe Anfragesprache verwenden

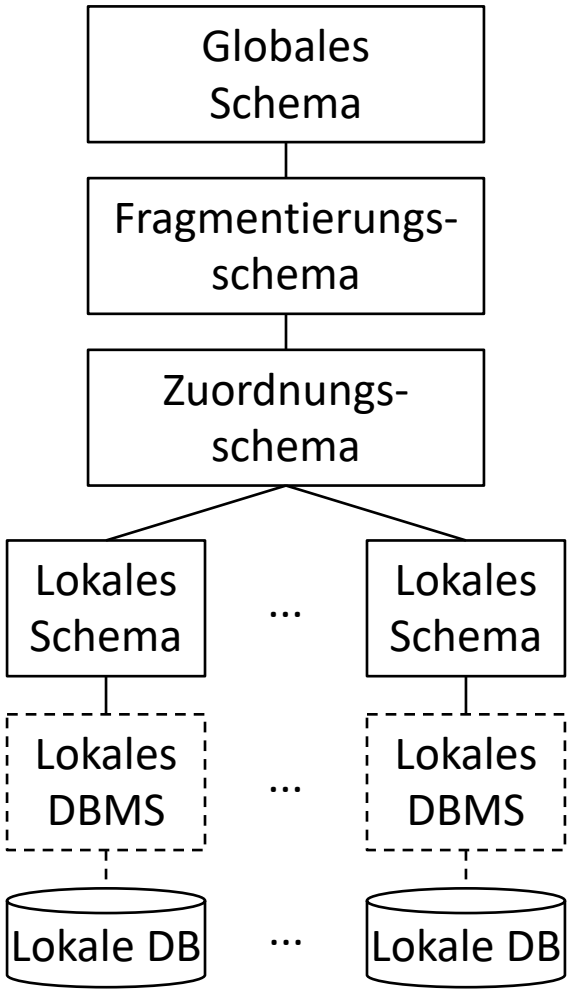


Abbildung nach Kemper & Eickler

Transaktionskontrolle in VDBMS

- Herausforderungen bei write_item(X)-Operationen
  1. X liegt über mehrere Fragmente und Stationen verteilt vor
     - Entsprechende Änderung muss für jedes Fragment/auf jeder Station realisiert werden
  2. X liegt repliziert vor
     - Änderung muss auf allen Replikaten realisiert werden, sonst inkonsistente Daten
  - Kann auch beides auf X zutreffen

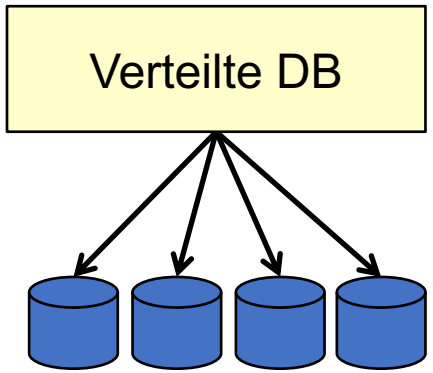
1. Transaktionsverwaltung muss verschiedene Änderungen über verschiedene Stationen hinweg schreiben (durch Fragmentierung)
   - Lokale Transaktionsverwaltung muss entsprechende Sperren anfordern
   - Verklemmungen können über Stationen hinweg entstehen (Erkennung oder Vermeidung noch aufwendiger)
2. Transaktionsverwaltung muss die gleichen Änderungen auf verschiedene Stationen schreiben (durch Replikation)
   - Lokale Transaktionsverwaltung muss entsprechende Sperren anfordern
   - Bei Abbruch an einer Station, Abbruch auf allen Stationen!
     - Zwei-Phasen-Commit-Protokoll (PREPARE, READY/FAILED, COMMIT/ABORT, ACK)

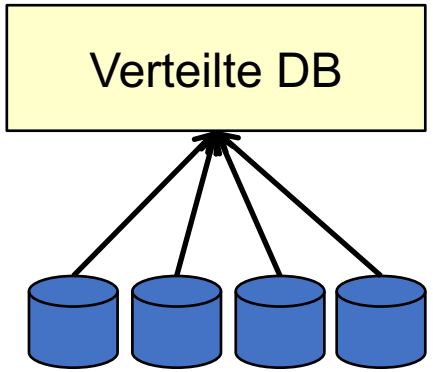
Zwischen-Rückblick

- Motivation: Verteilung und Dezentralisierung
- Fragmentierung
  - Horizontal
  - Vertikal
  - Abgeleitet
- Replikation
- Allokation
- Transparenz
  - Fragmentierungstransparenz
  - Allokationstransparenz
  - Lokale-Schema-Transparenz

→ Davon abhängig: Aufwand für das VDBMS

  - Anfragebearbeitung
  - Änderungsoperationen
  - Damit einhergehend: Transaktionskontrolle





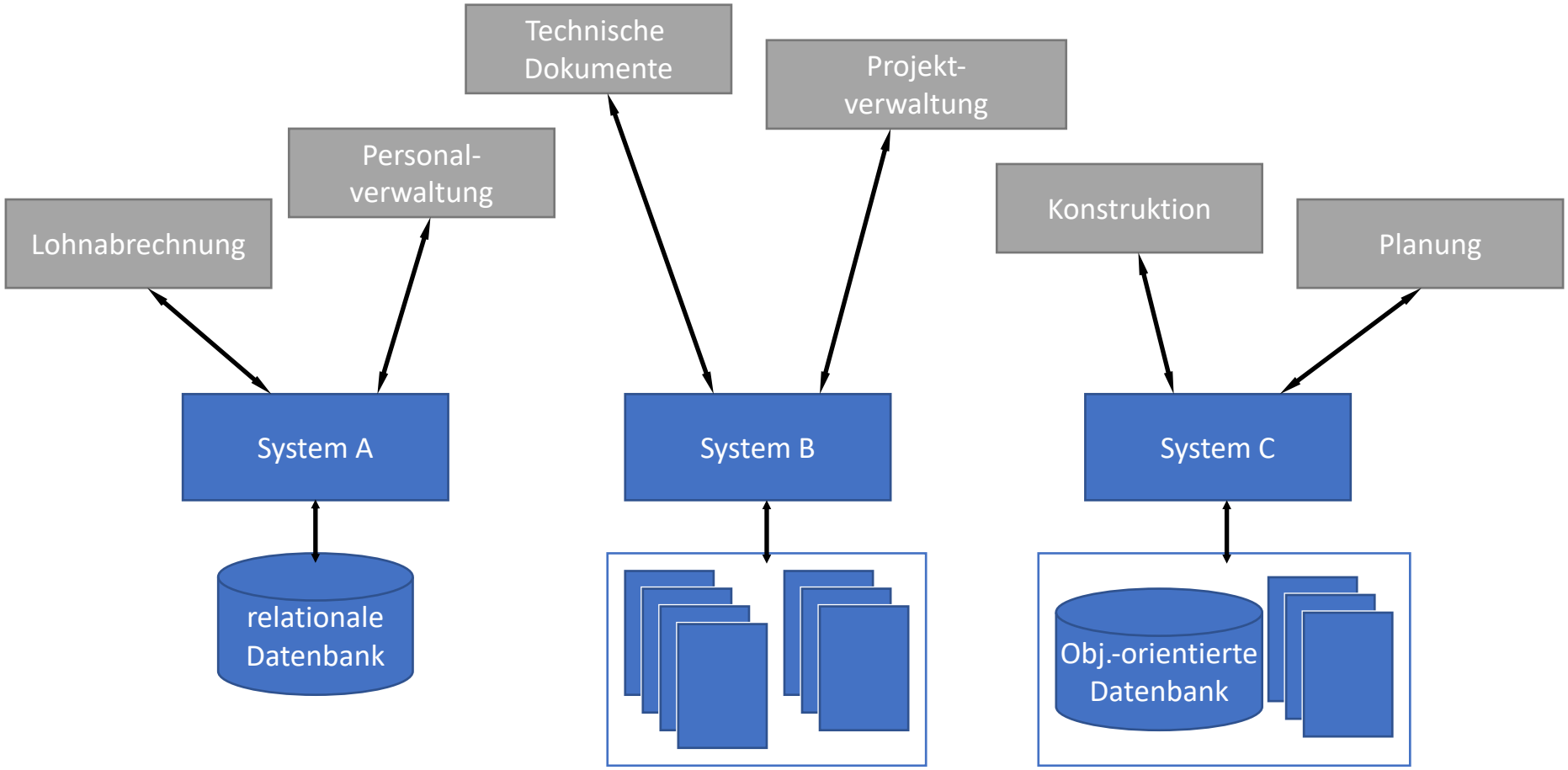
Föderierte DBS

Integration und Migration von DBs

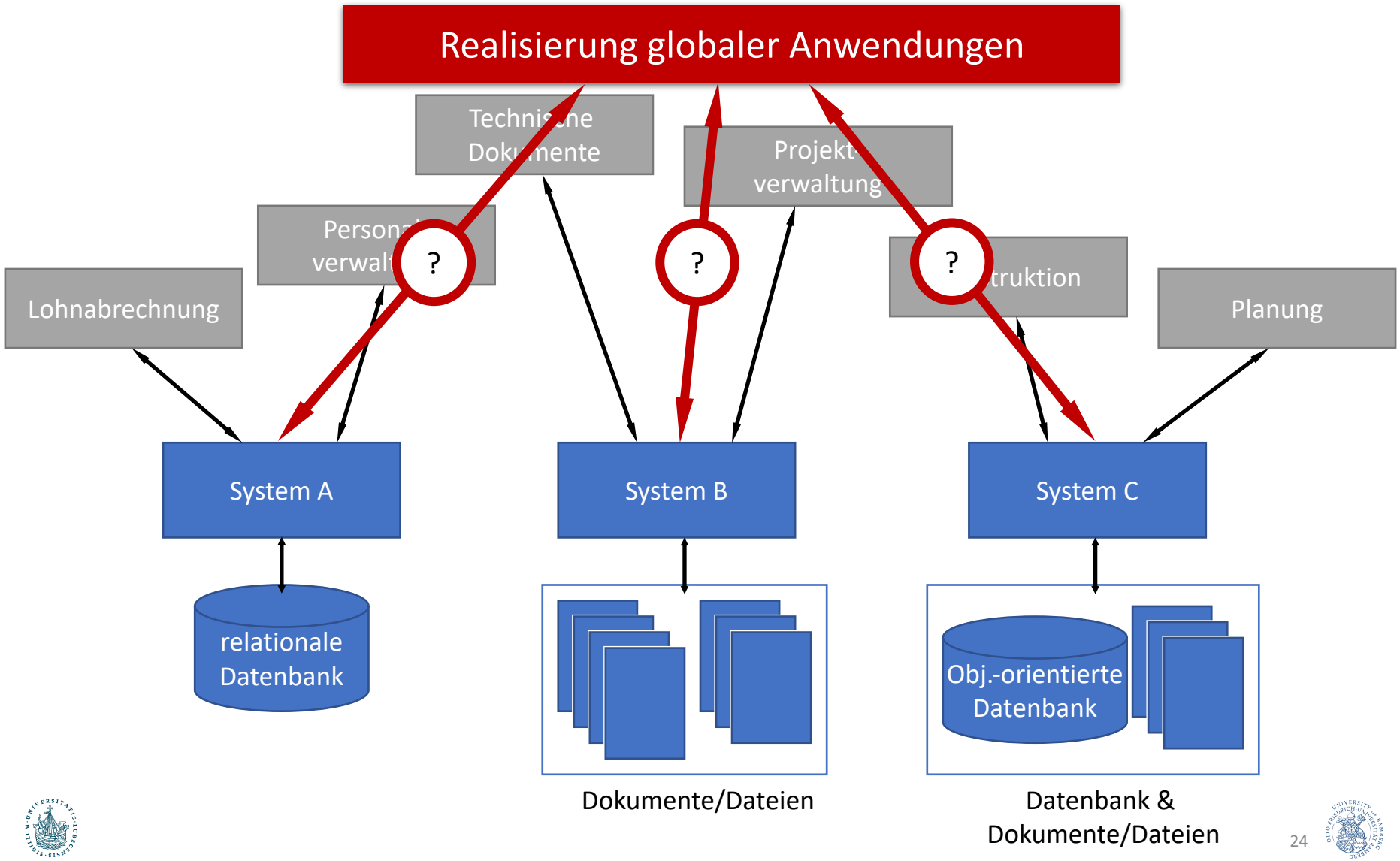
Anwendungsbeispiel

- Typische Situation in Unternehmen und Organisationen:
  - Unabhängig voneinander entstandene Datenbestände in verschiedenen Abteilungen/Arbeitsgruppen
    - Unterschiedliche Strukturierung gleich(artig)er Daten
    - Teilweise redundante Datenhaltung
    - Global inkonsistente Datensituation möglich
  - Autonome, nicht abgestimmte Entscheidungen über Anschaffung von Hardware und Software
- Entstehung von „Insellösungen“, die einen einheitlichen Zugriff auf verteilte Daten erschweren oder sogar unterbinden

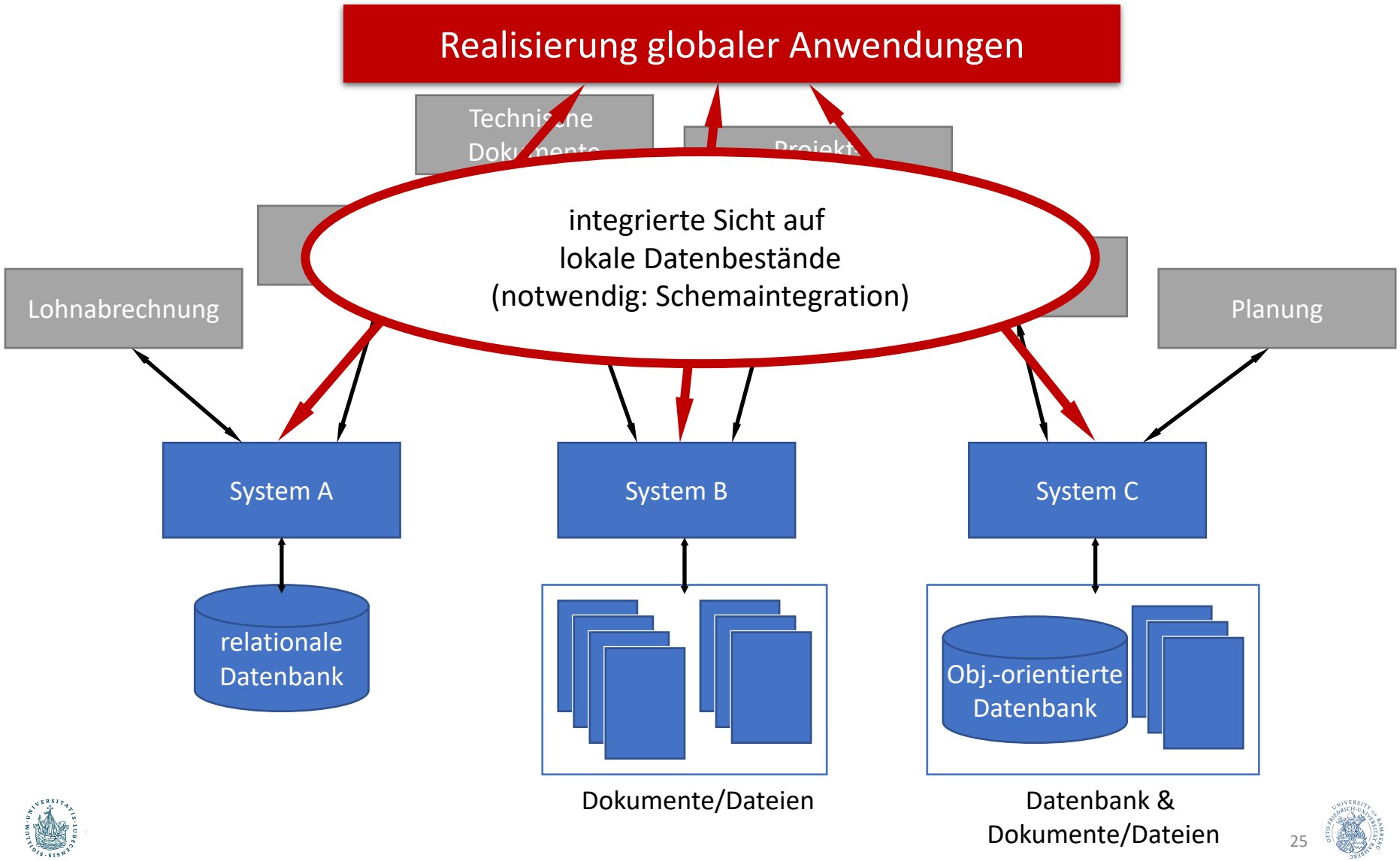
Anwendungsbeispiel



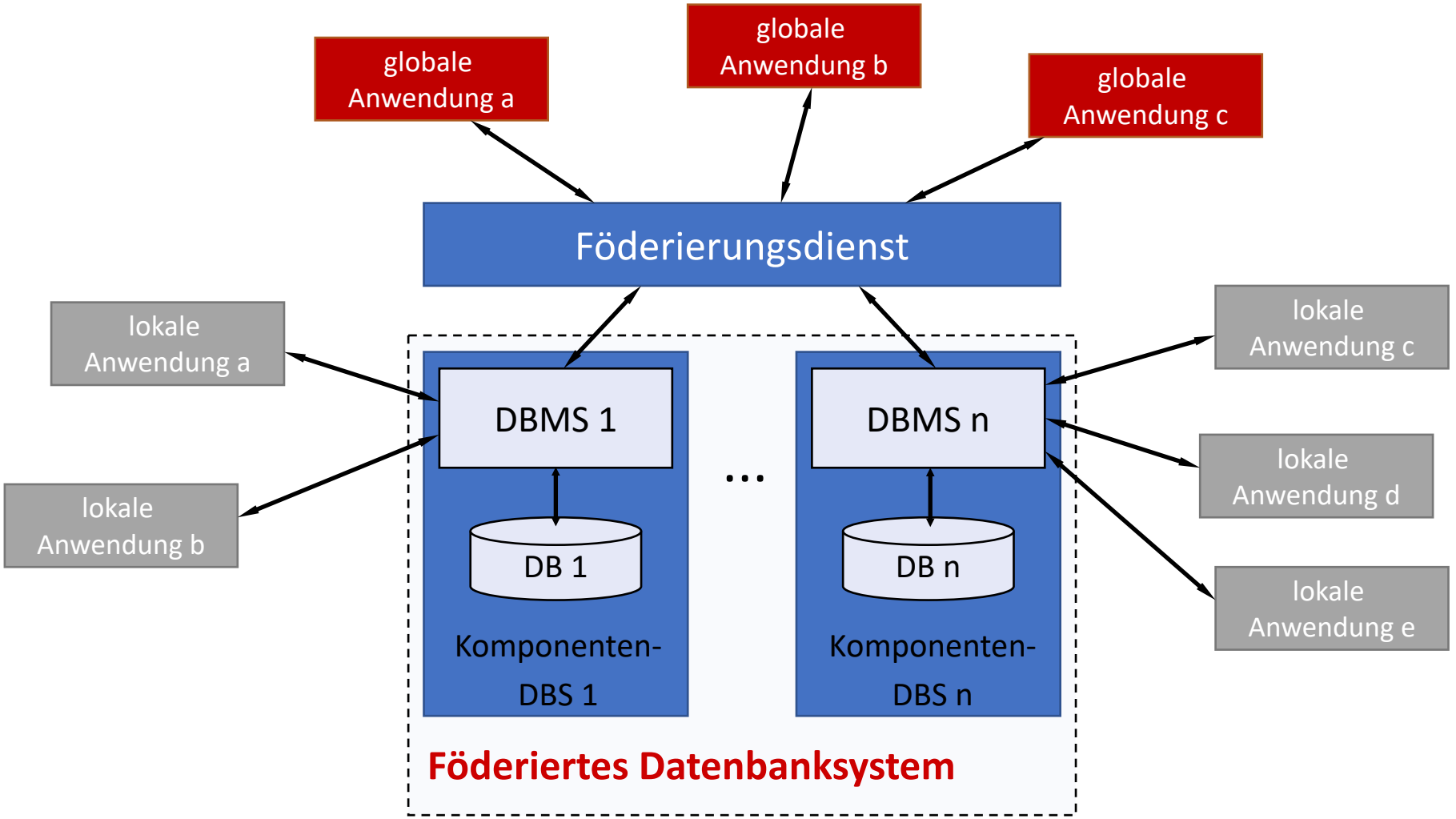
Anwendungsbeispiel



Anwendungsbeispiel



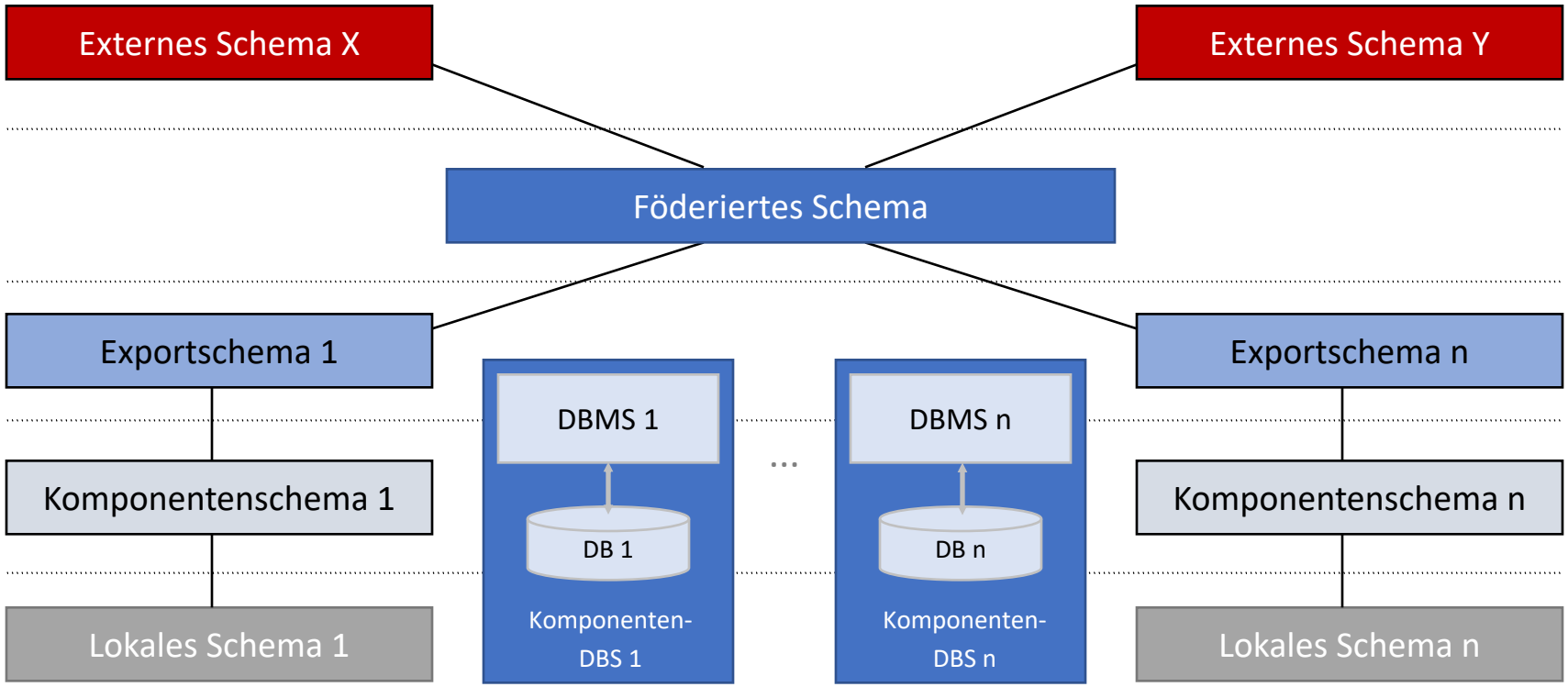
Föderierte DBS– Allgemeine Architektur



Was soll ein föderiertes DBS können?

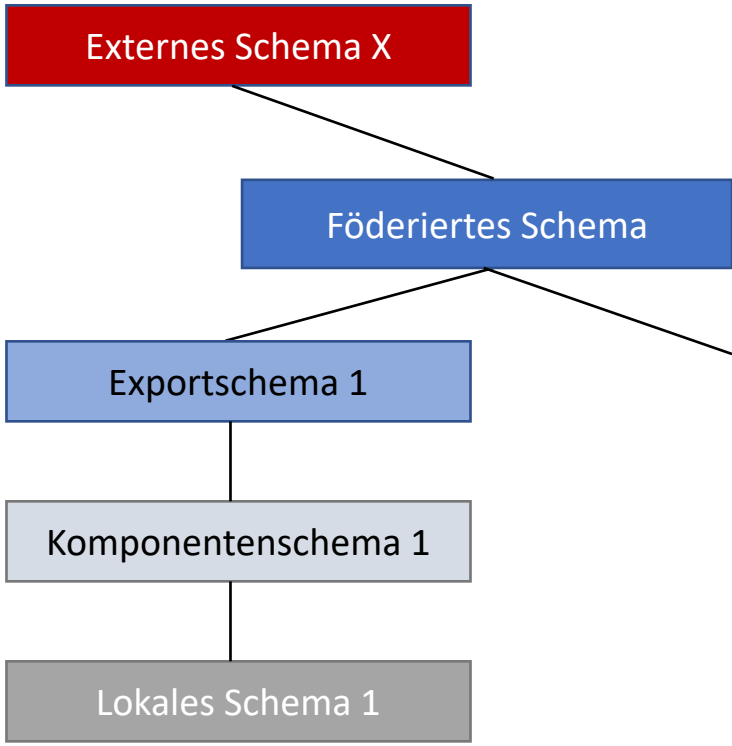
- Im Idealfall
  - Volle Funktionalität eines DBS, Verbergen der Verteilung bzgl.
    - Transaktionsverwaltung
    - Integritätskontrolle
    - Anfrageverarbeitung und -optimierung
    - Recovery
    - ...
  - ... auf globaler Ebene (im Föderierungsdienst)!
  - Bedeutet insgesamt hohen Aufwand
- Pragmatischer Ansatz:
  - Nur solche Funktionalität im föderierten System realisieren, die tatsächlich benötigt wird – eine stark anwendungsgetriebene Diskussion

5-Schichten-Architektur



Schemaintegration

- Lokales Schema
  - Datenschema des lokalen DBS
- Komponentenschema
  - Schema des lokalen DBS im globalen Datenmodell
- Exportschema
  - Teile des lokalen Schemas, die in Föderation eingebracht werden sollen
- Föderiertes Schema
  - Globales Schema des föderierten DBS
- Externes Schema
  - Schema für einzelne Anwendungen bzw. Benutzer



Schemaintegration

Vorgehen

- Schematransformation:
  - Übersetzung der lokalen Schemata in Komponentenschemata (semi-automatisch oder interaktiv anhand von Transformationsregeln)
- Verhandlung (unter Domänenexperten und Datenbankdesignern):
  - Anwendungsorientierte Diskussion der angestrebten Exportschemata
- Eigentliche Schemaintegration:
  - Vereinigung der Exportschemata zu föderiertem Schema

Probleme

- Synonyme: gleiche Bedeutung, aber unterschiedliche Benennung
- Homonyme: gleiche Namen, aber unterschiedliche Bedeutung
- Ähnlichkeiten, Spezialisierungen, Überschneidungen

Schemaintegration → Datenintegration

	Semantische Ebene	Zu behebende Inkonsistenz	Angewandte Methoden
1	Schema	Schematische Heterogenität	Integration Mapping Matching
2	Tupel (Object)	Duplikate	Ähnlichkeitsmaße Partitionierungs-strategien
3	Wert	Datenkonflikte	Datenreinigung Transformationen Fusion

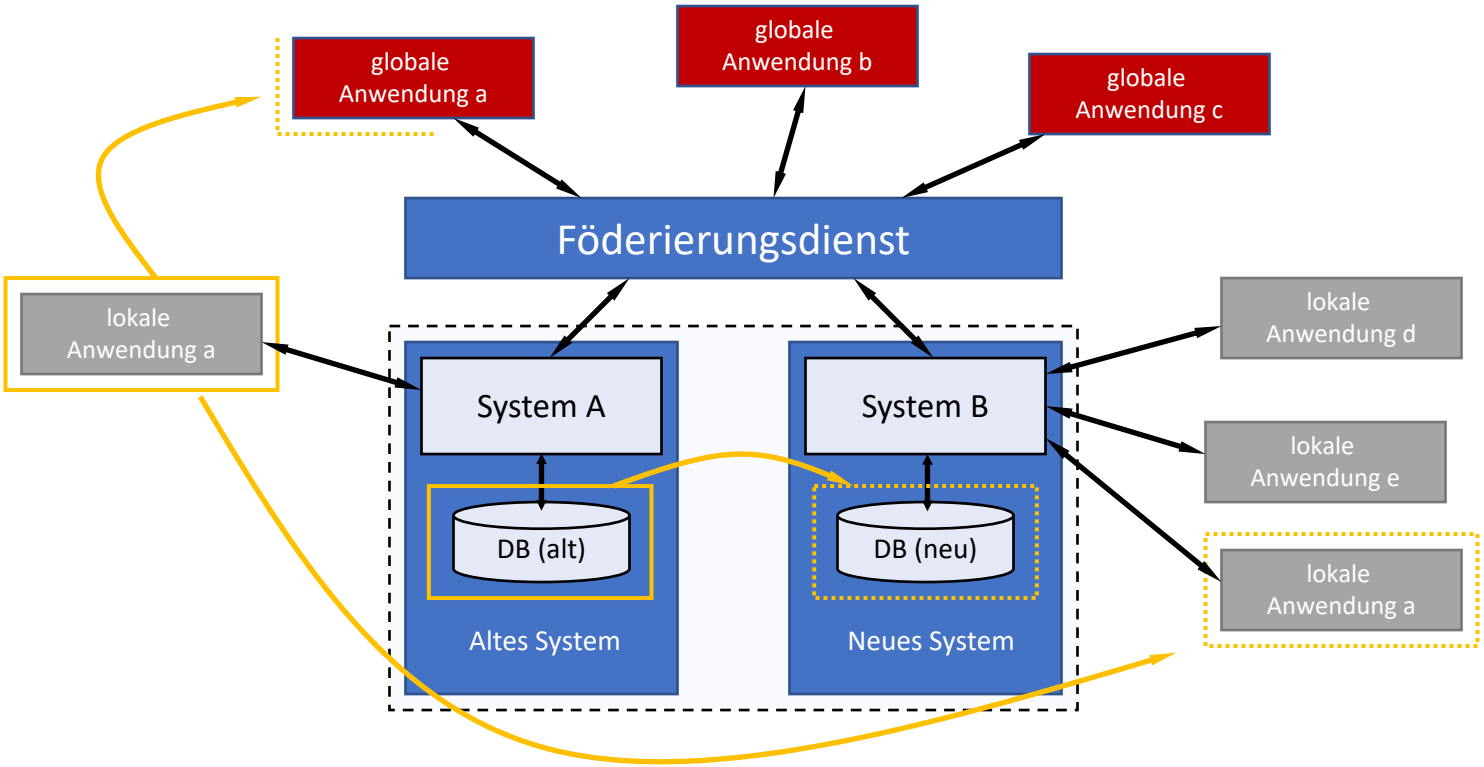
Beispiele für föderierte DBS

- Ablösung von Altsystemen (legacy systems) – z.B. wegen ...
  - Ansteigender Wartungskosten
  - Nachlassender Unterstützung durch Systemhersteller
  - Überlastung des Altsystems in vorliegender bzw. perspektivischer Anwendungssituation – keine ausreichenden Skalierungsmöglichkeiten
  - Reorganisation betrieblicher Strukturen – z.B. aufgrund einer Fusionen mit anderem Unternehmen, die letztlich die Vereinheitlichung der Systemplattformen notwendig macht
- Wichtige Anforderungen dabei
  - Investitionsschutz
    - Erhalt von Daten
    - Erhalt von Anwendungsprogrammen
  - Fortlaufender Betrieb während Migration

Migration in kleinen Schritten

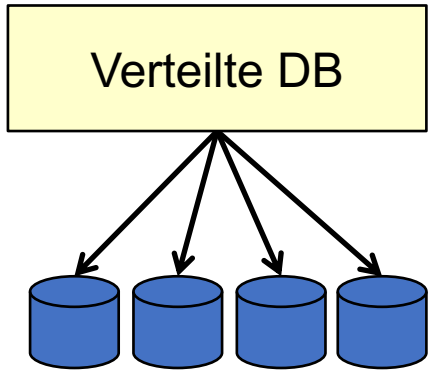
Für fortlaufenden Betrieb

- System B aufsetzen, Föderierungsdienst definieren
- Lokale Anwendung a in globale Anwendung a übersetzen
- Daten aus System A übertragen
- Lokale Anwendung a an System B überführen, dann alte lokale Anwendung a und globale Anwendung a löschen



Zusammenfassung

- Verteilte Datenbanken
  - Verteilung einer DB auf Stationen
  - Fragmentierung
  - Replikation
  - Allokation
  - Transparenz
  - Herausforderungen der Transaktionskontrolle



- Föderierte Datenbanken
  - Einheitliche Sicht auf unterschiedliche DBs
  - Integration
  - Systematische Migration
  - ... von Daten und Anwendungen