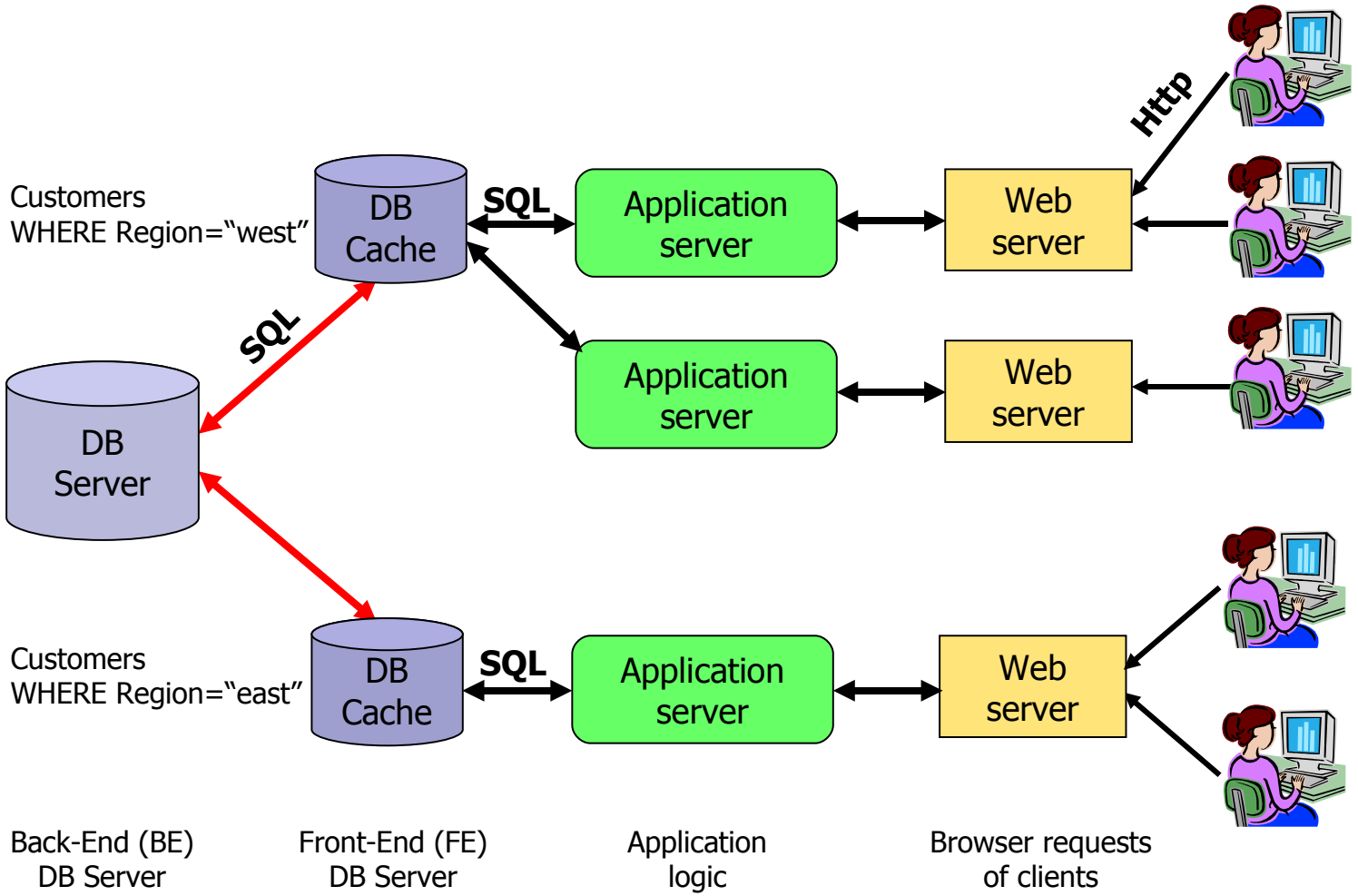
TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Wer trägt den Müll raus?

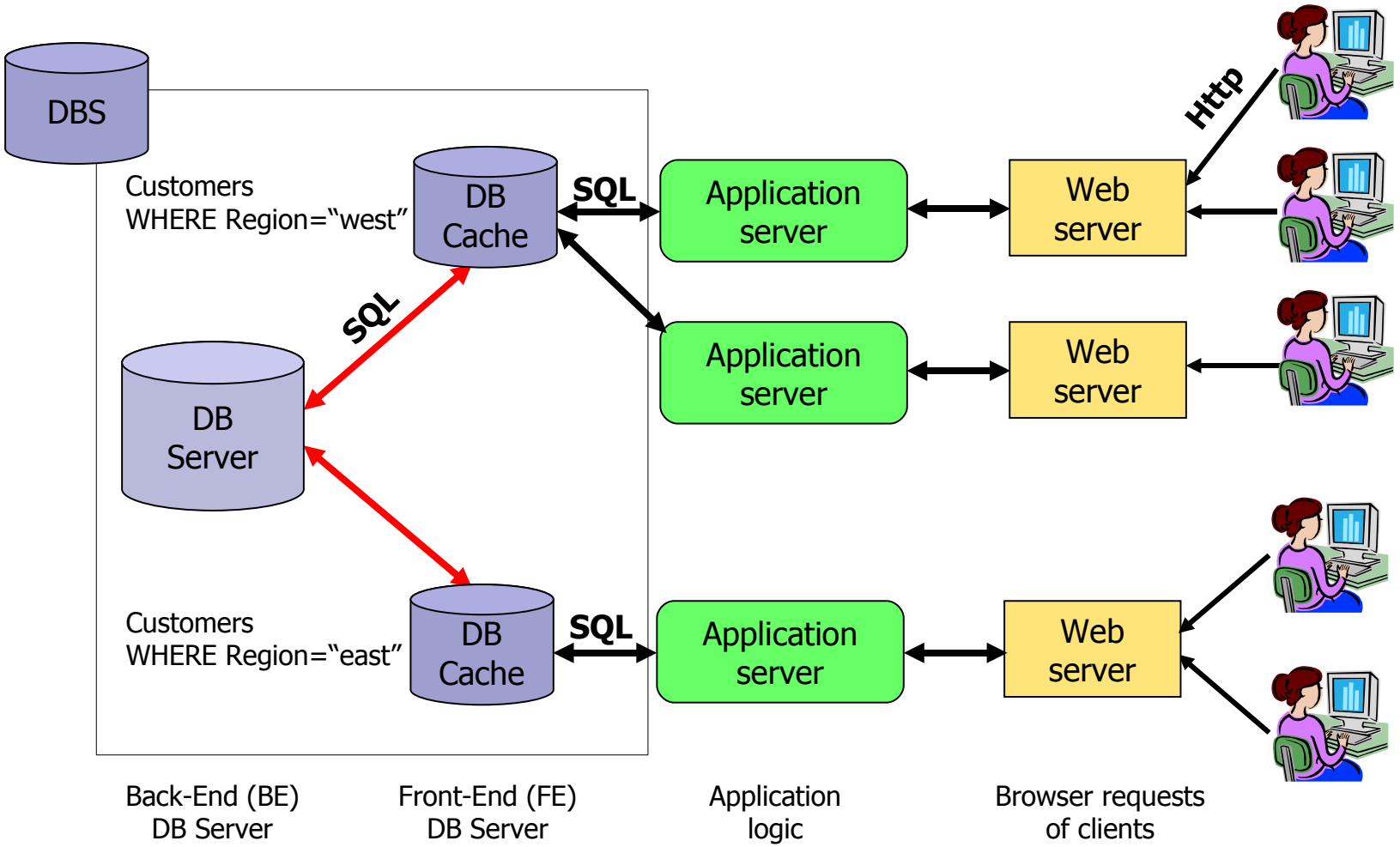
Verteilung der Hauptaufgaben im Datenbank-Caching (CbDBC)

Joachim Klein, 26.09.2008

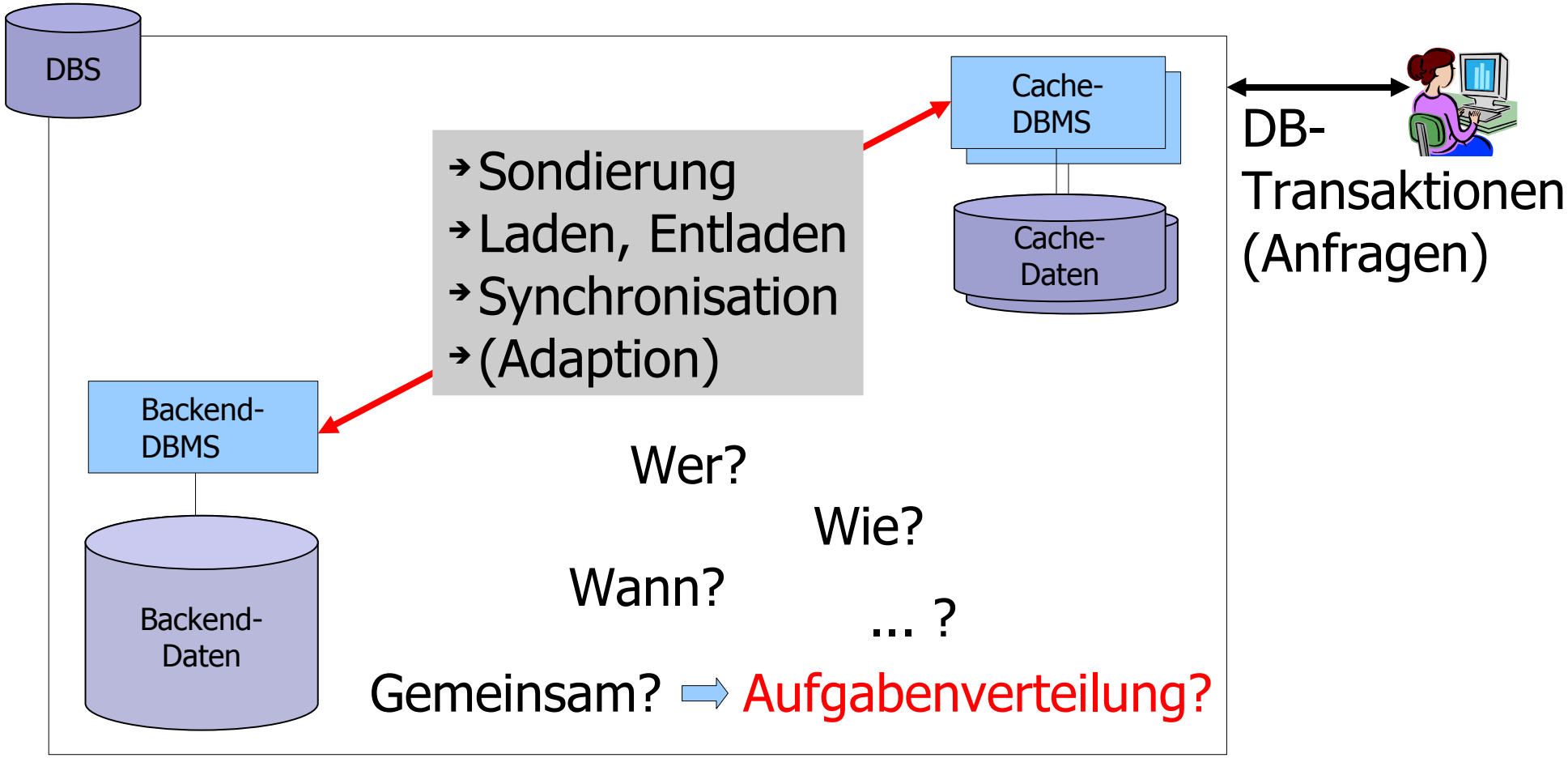
Datenbank-Caching



Datenbank-Caching



Hauptaufgaben



Funktionsprinzip
(Constraint-basiertes Datenbank-Caching)

Backend-Datenbank

Abt.	Id	Name	Gebäude
	01	GBIS	36
	02	HIS	36

Person	Id	Name	Ort	Abt
	1	Andreas	KL	01
	2	Thomas	MZ	02
	3	Jürgen	KL	02
	4	Joachim	ZW	01

```
SELECT p.Id, p.Name
FROM Person, Abt
WHERE p.Abt=a.Id
AND a.Name='GBIS'
order by p.Name
```

Funktionsprinzip
(Constraint-basiertes Datenbank-Caching)

```
SELECT p.Id, p.Name
FROM Person, Abt
WHERE p.Abt=a.Id
AND a.Name='GBIS'
order by p.Name
```

Backend-Datenbank

Abt.	Id	Name	Gebäude
	01	GBIS	36
	02	HIS	36

Person	Id	Name	Ort	Abt
	1	Andreas	KL	01
	2	Thomas	MZ	02
	3	Jürgen	KL	02
	4	Joachim	ZW	01

Cache-Datenbank

C_Abt.	Id	Name	Gebäude
	01	GBIS	36

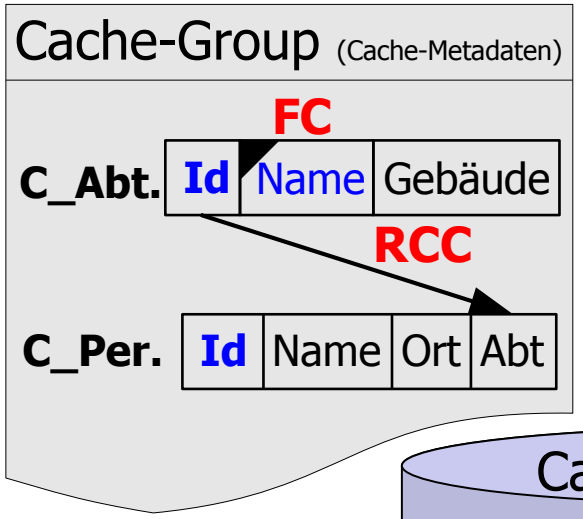
C_Per.	Id	Name	Ort	Abt
	1	Andreas	KL	01
	4	Joachim	ZW	01

Funktionsprinzip
(Constraint-basiertes Datenbank-Caching)

Backend-Datenbank

Abt.	Id	Name	Gebäude
	01	GBIS	36
	02	HIS	36

Person	Id	Name	Ort	Abt
	1	Andreas	KL	01
	2	Thomas	MZ	02
	3	Jürgen	KL	02
	4	Joachim	ZW	01




```
SELECT p.Id, p.Name
FROM Person, Abt
WHERE p.Abt=a.Id
AND a.Name='GBIS'
order by p.Name
```

Cache-Datenbank

C_Abt.	Id	Name	Gebäude
	01	GBIS	36

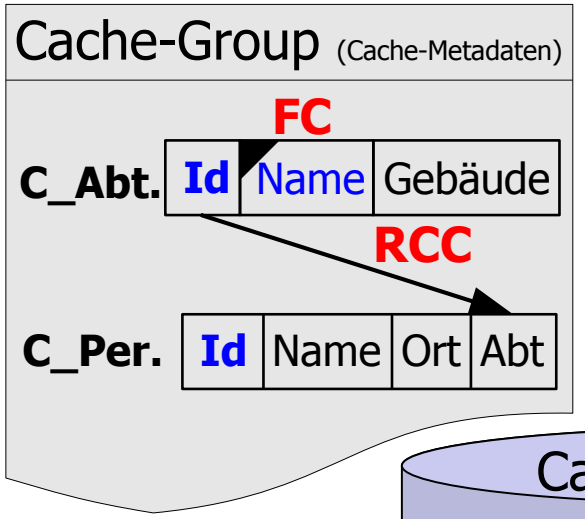
C_Per.	Id	Name	Ort	Abt
	1	Andreas	KL	01
	4	Joachim	ZW	01

Funktionsprinzip
(Constraint-basiertes Datenbank-Caching)

Backend-Datenbank

Abt.	Id	Name	Gebäude
	01	GBIS	36
	02	HIS	36

Person	Id	Name	Ort	Abt
	1	Andreas	KL	01
	2	Thomas	MZ	02
	3	Jürgen	KL	02
	4	Joachim	ZW	01




```
SELECT p.Id, p.Name
FROM Person, Abt
WHERE p.Abt=a.Id
AND a.Name='GBIS'
order by p.Name
```

Cache-Datenbank

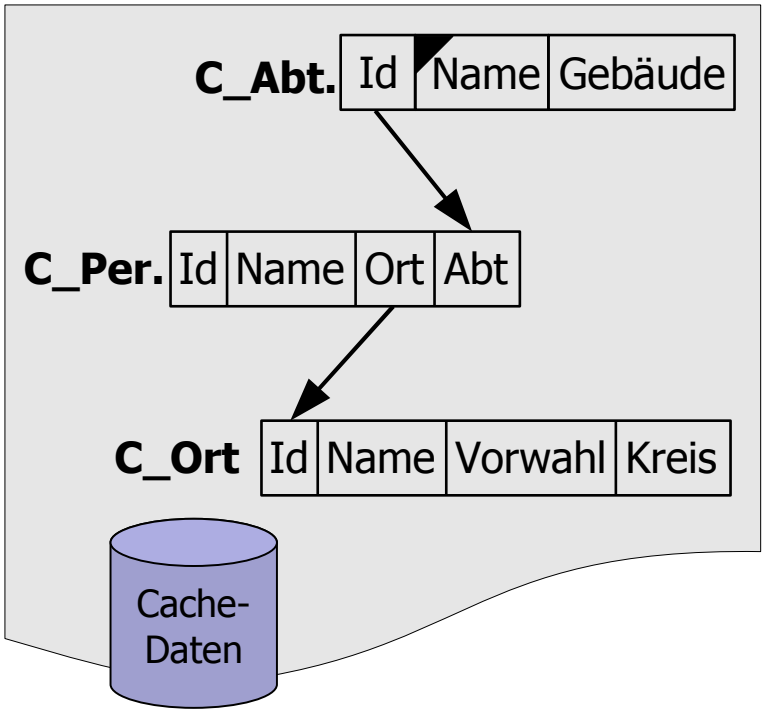
C_Abt.	Id	Name	Gebäude
	01	GBIS	36

Cache-Unit „GBIS“

C_Per.	Id	Name	Ort	Abt
	1	Andreas	KL	01
	4	Joachim	ZW	01

Problem: Verteilung

Beispiel: Ladevorgang

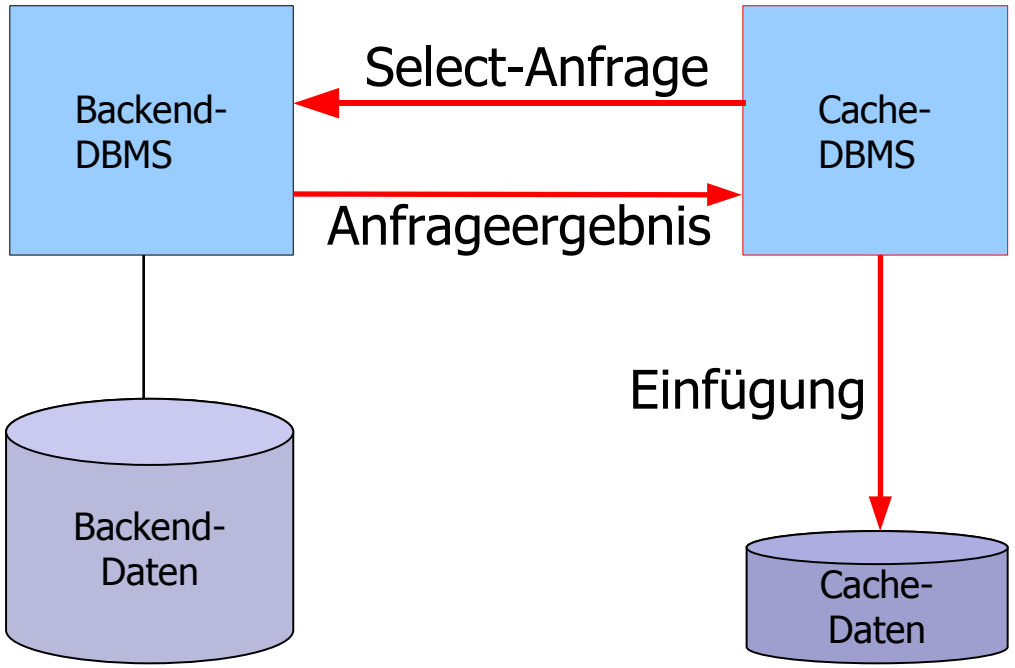


Lade: Abt.Name="GBIS"

Einfüllen der Daten sollte Bottom-Up erfolgen, da sonst RCCs verletzt werden.

Problem: Verteilung

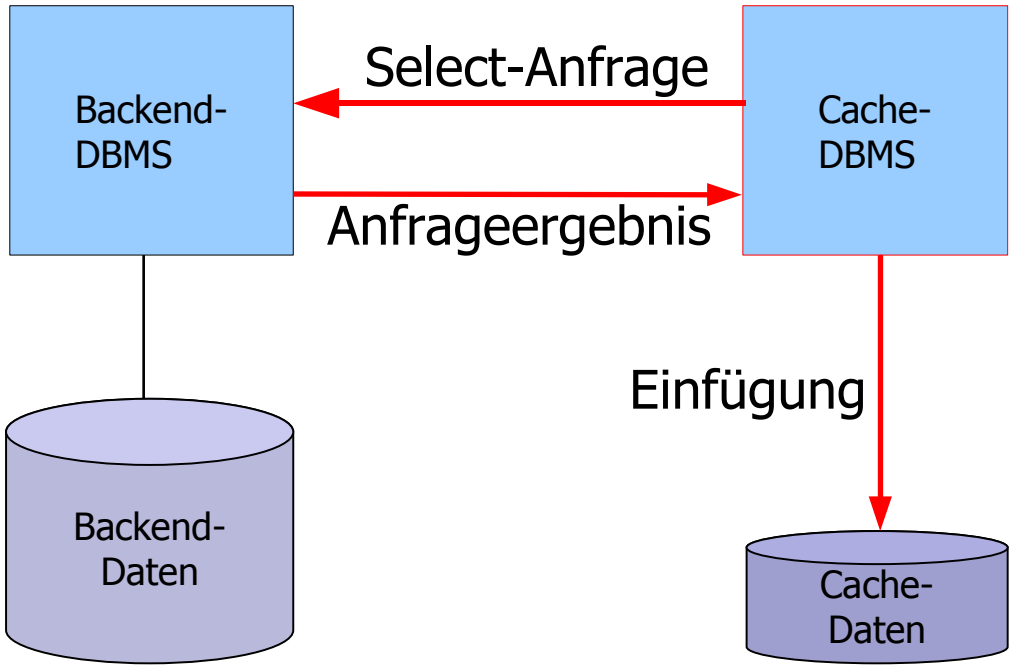
Direktes Laden



- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

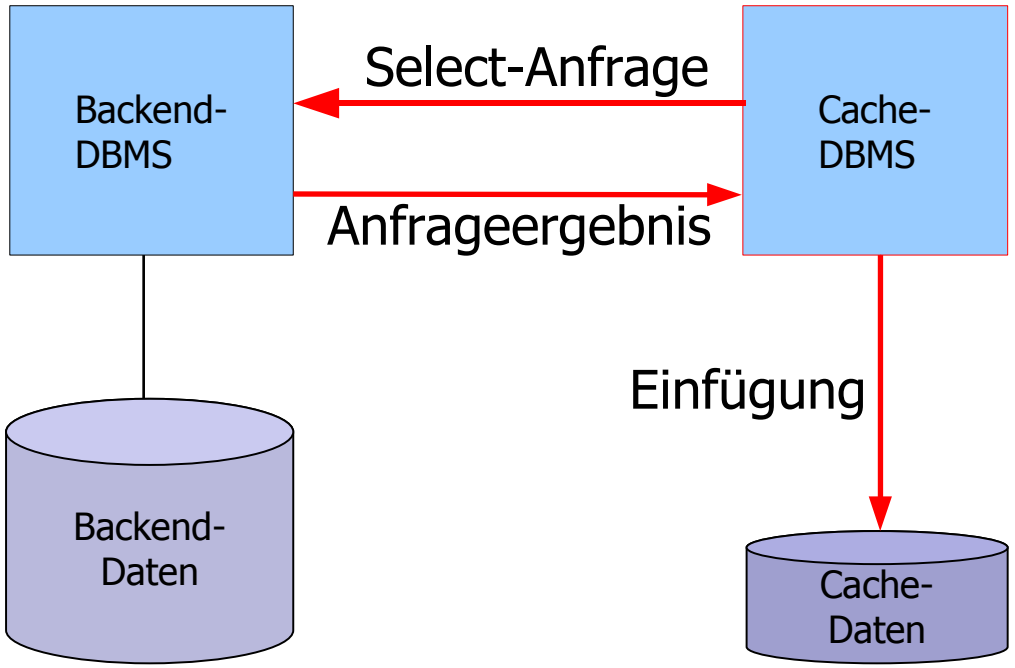


✔ Direktes Einfüllen der Anfrageergebnisse

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

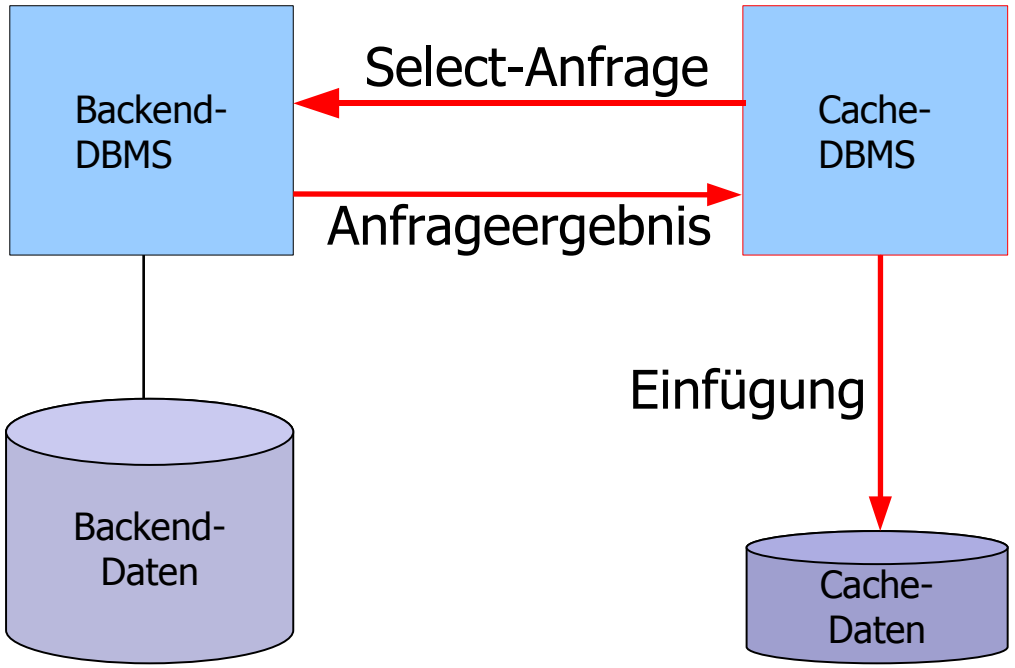


- ✔ Direktes Einfüllen der Anfrageergebnisse
- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

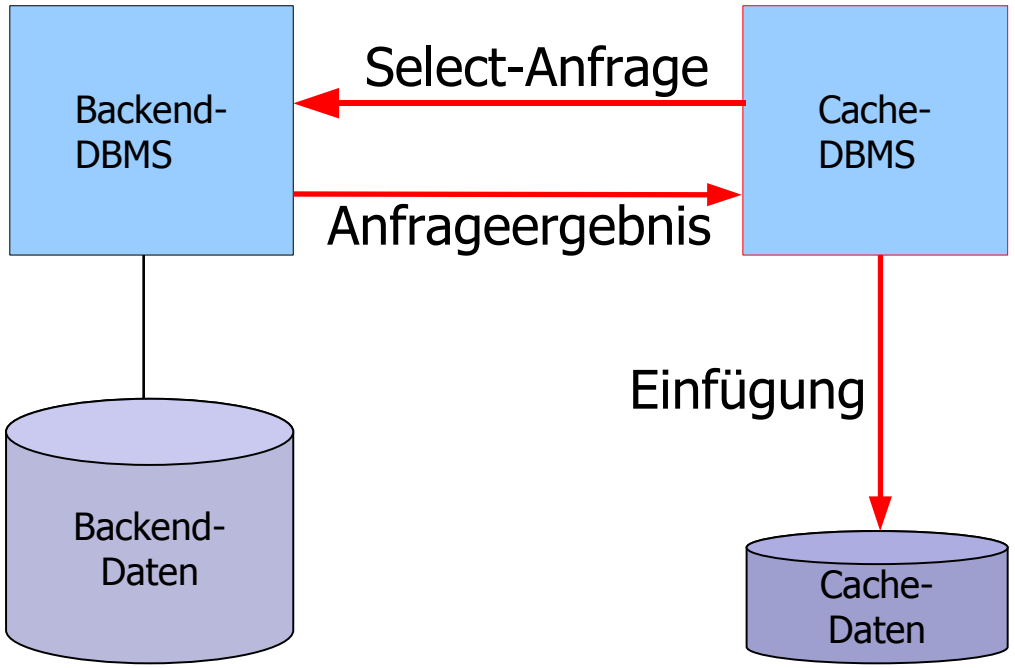


- ✔ Direktes Einfüllen der Anfrageergebnisse
- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)
- ✔ Backend benötigt keinerlei Cache-Metadaten

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

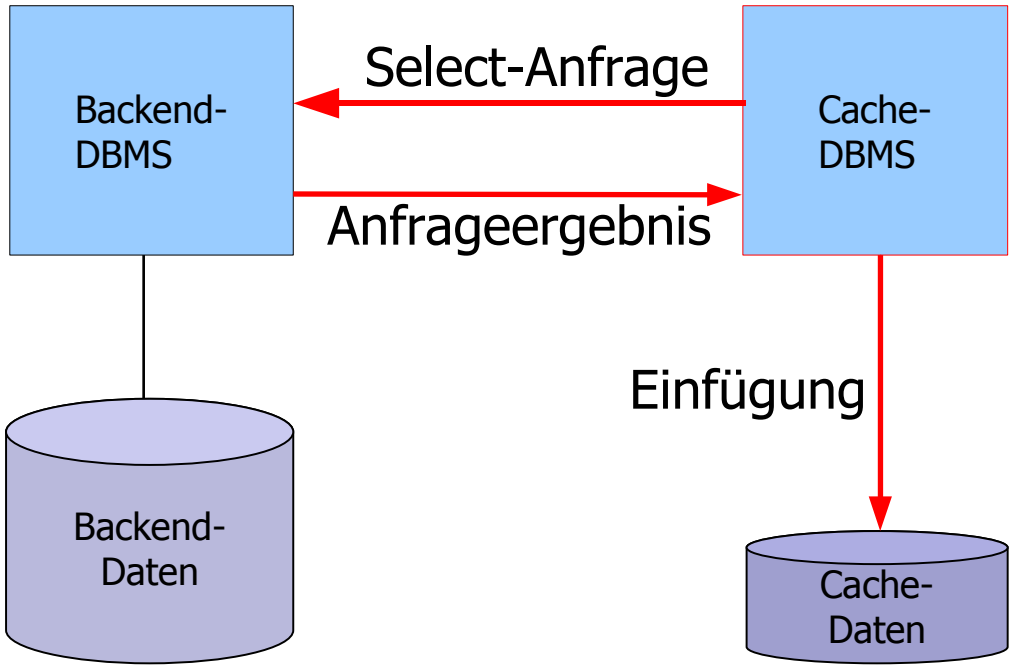
Direktes Laden



- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

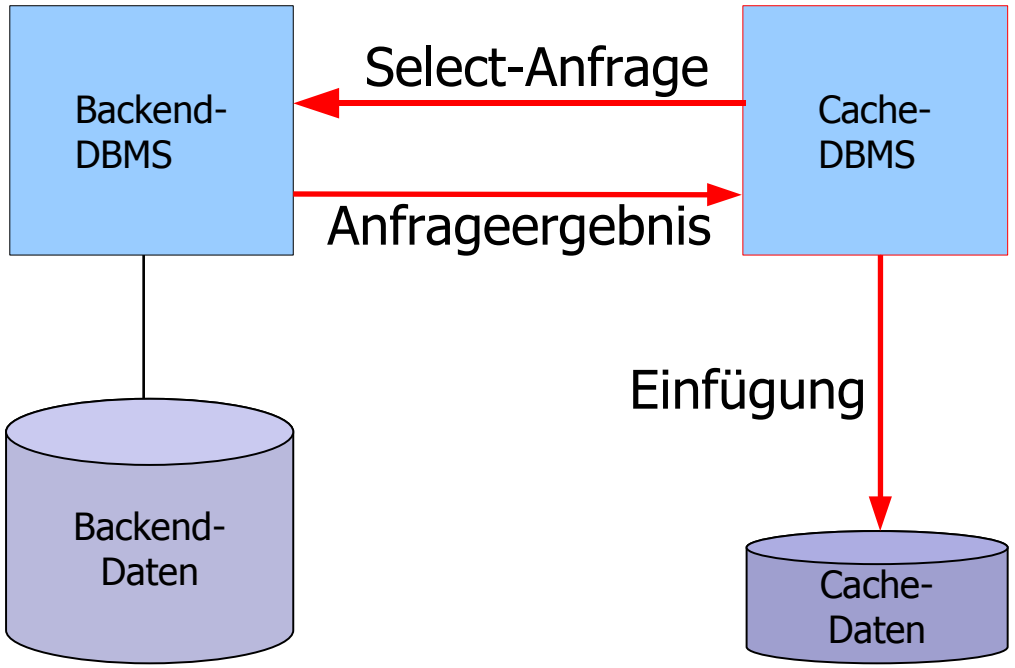


✗ Mehrfaches Anfragen vom Cache an Backend
➔ Latenzproblematik

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

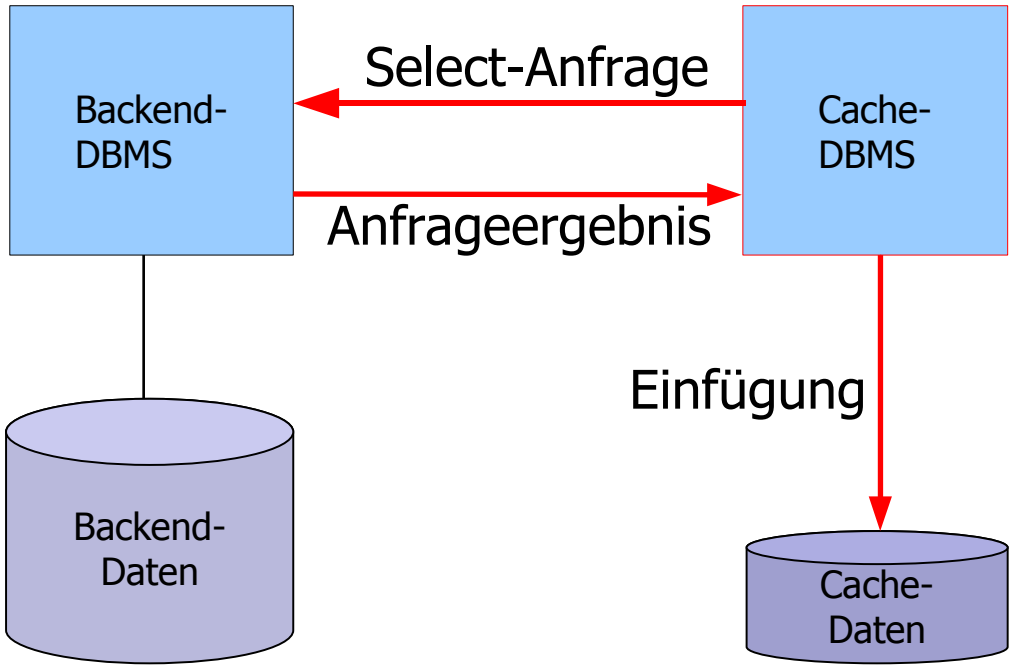


- ✗ Mehrfaches Anfragen vom Cache an Backend
  ➔ Latenzproblematik
- ✗ Große Abhängigkeit vom Füllgrad der Cache-Tabellen

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

Problem: Verteilung

Direktes Laden

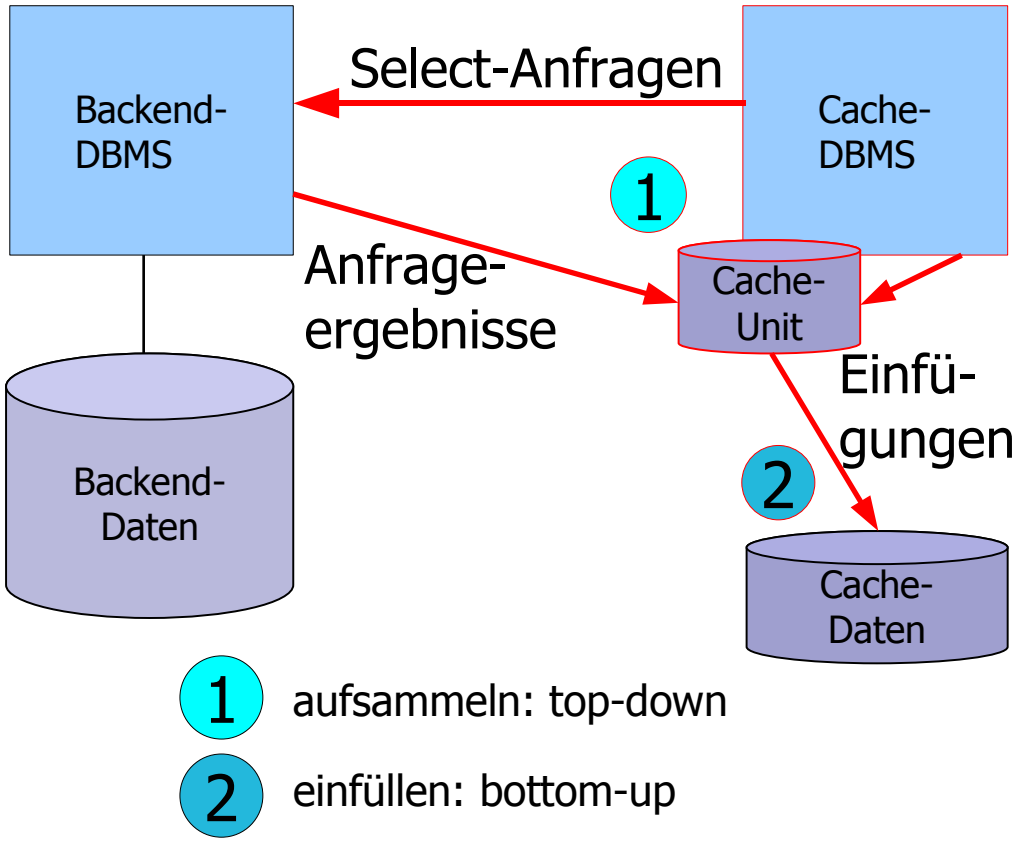


- ✗ Mehrfaches Anfragen vom Cache an Backend
  ➔ Latenzproblematik
- ✗ Große Abhängigkeit vom Füllgrad der Cache-Tabellen
- ✗ Unkontrollierbar lange Prädikate in Anfragen

- Ablauf:
  - nur schematisch
  - wird mindestens ein mal pro zu füllender Tabelle durchlaufen

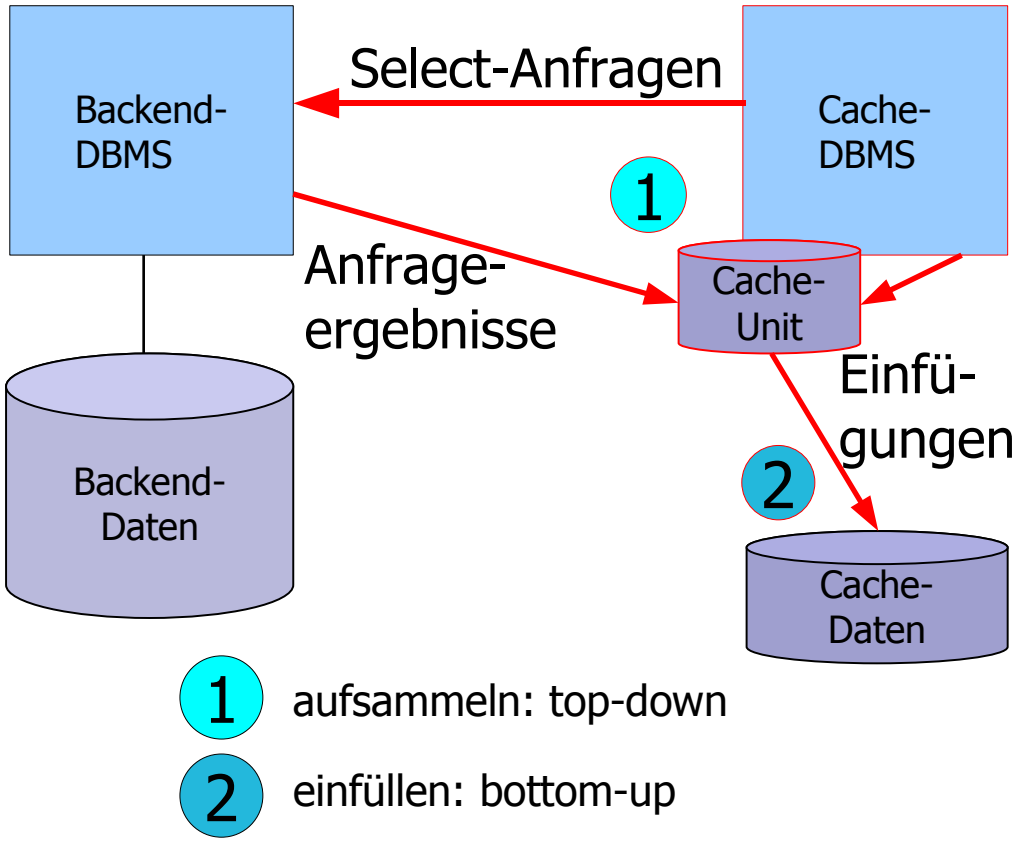
Problem: Verteilung

Indirektes Laden



Problem: Verteilung

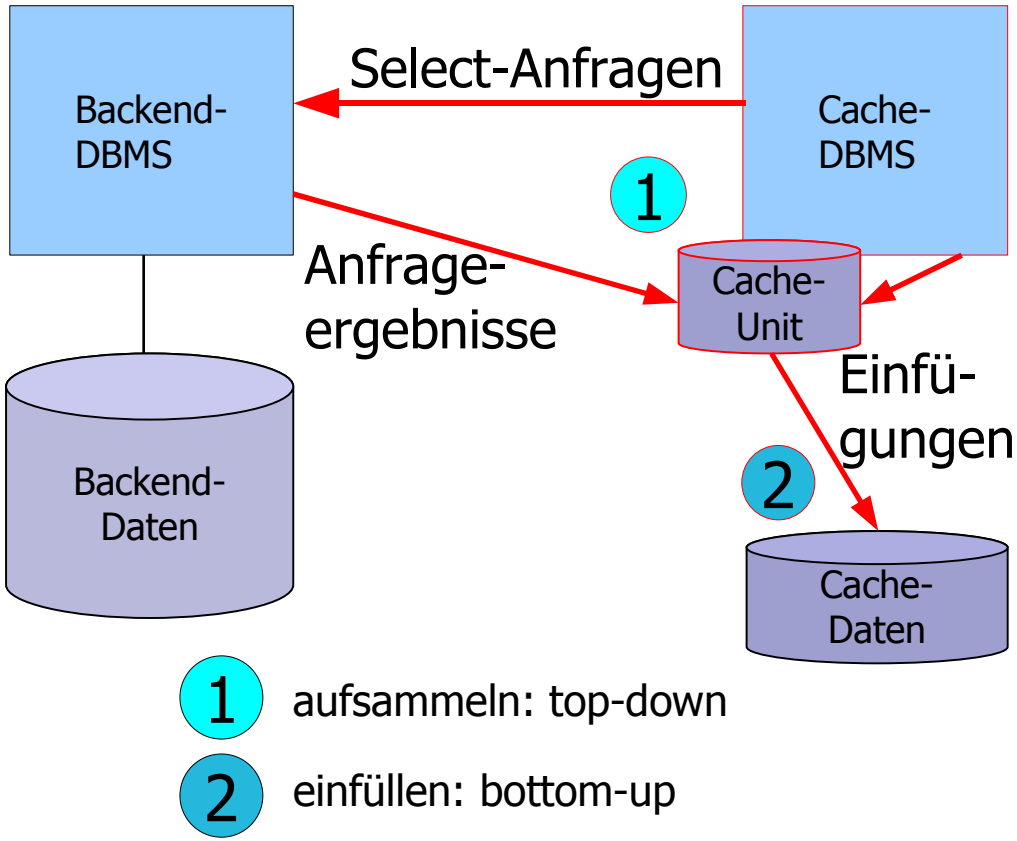
Indirektes Laden



- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)

Problem: Verteilung

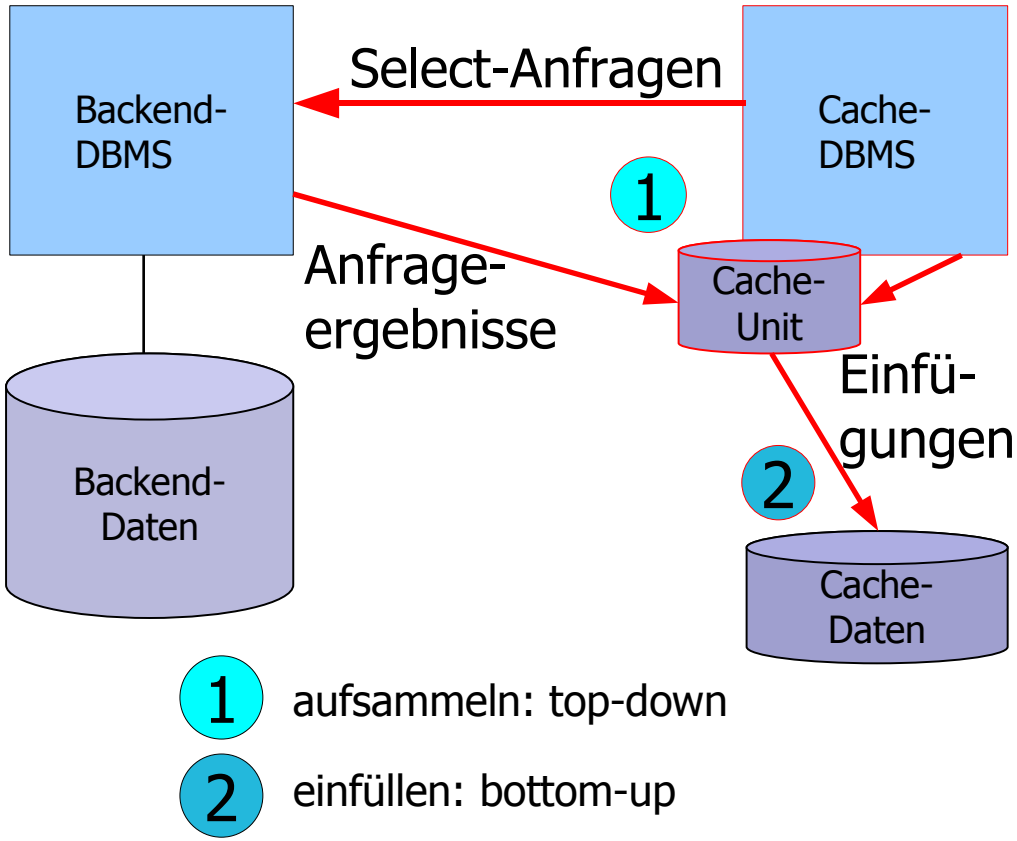
Indirektes Laden



- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)
- ✔ Backend benötigt keinerlei Cache-Metadaten

Problem: Verteilung

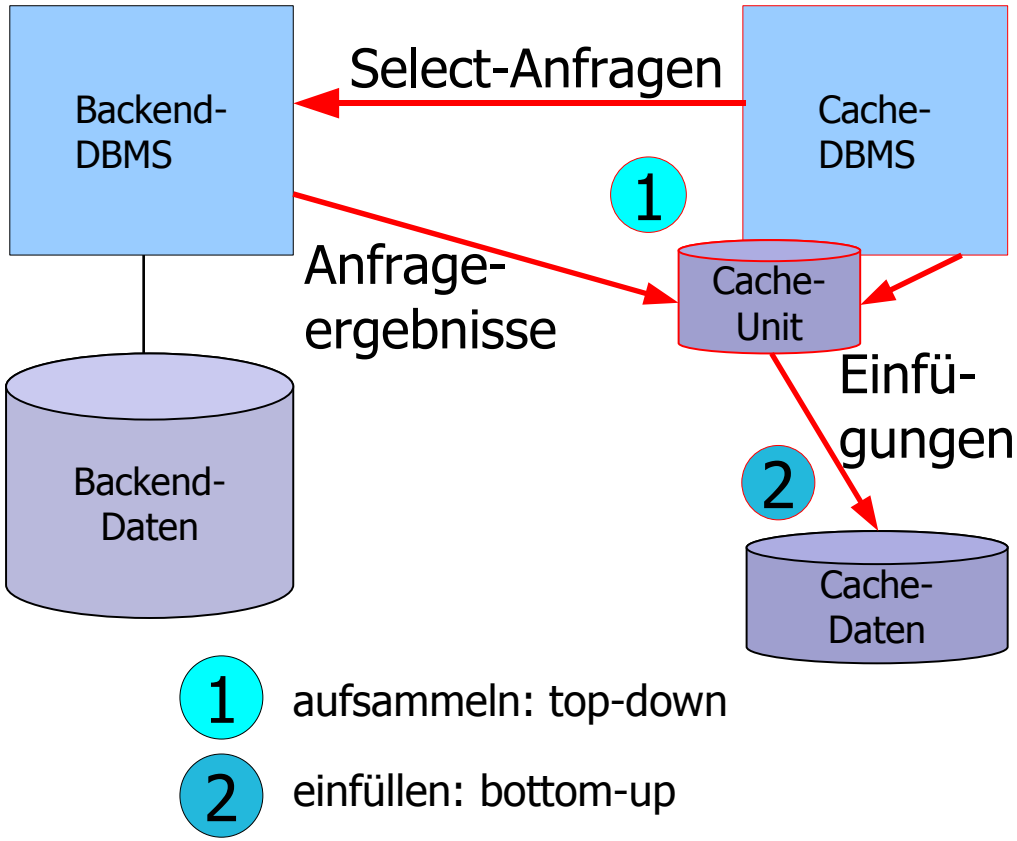
Indirektes Laden



- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)
- ✔ Backend benötigt keinerlei Cache-Metadaten
- ✔ Geringe Abhängigkeit vom Füllgrad der Cache-Tabellen

Problem: Verteilung

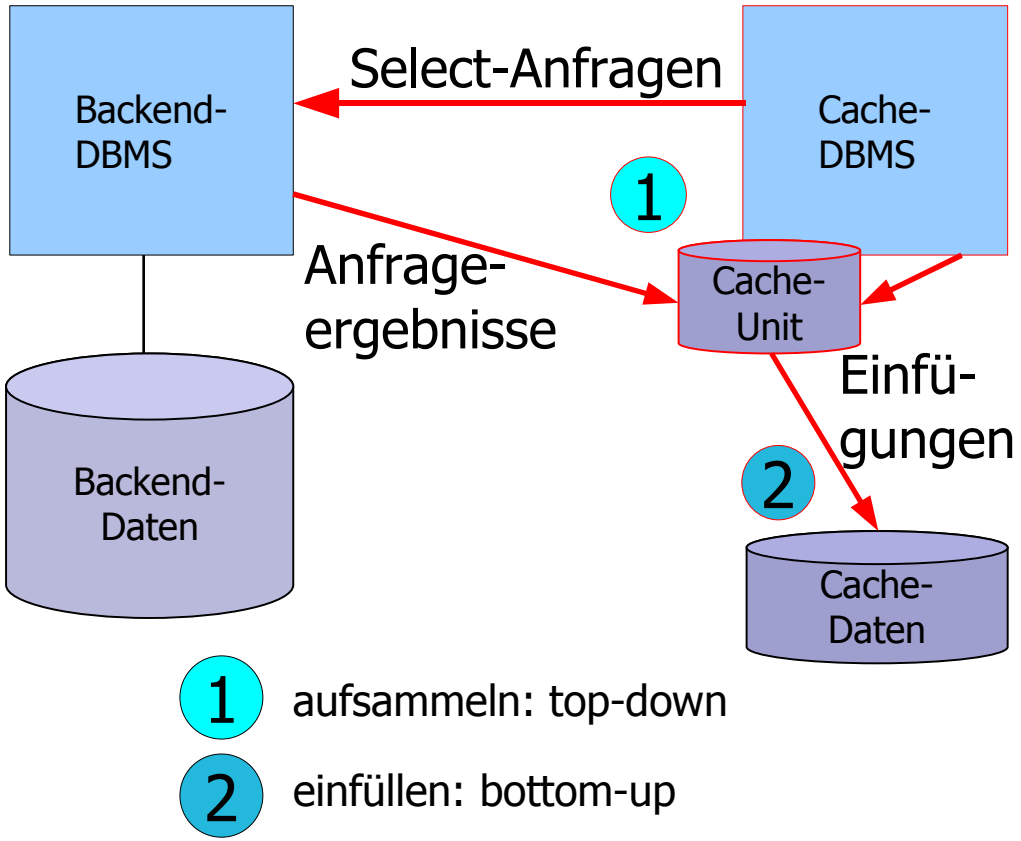
Indirektes Laden



- ✔ Cache-DBMS organisiert Ladevorgang (Backend-DBMS entlastet)
- ✔ Backend benötigt keinerlei Cache-Metadaten
- ✔ Geringe Abhängigkeit vom Füllgrad der Cache-Tabellen
- ✔ Konstante Prädikatlänge in Anfragen

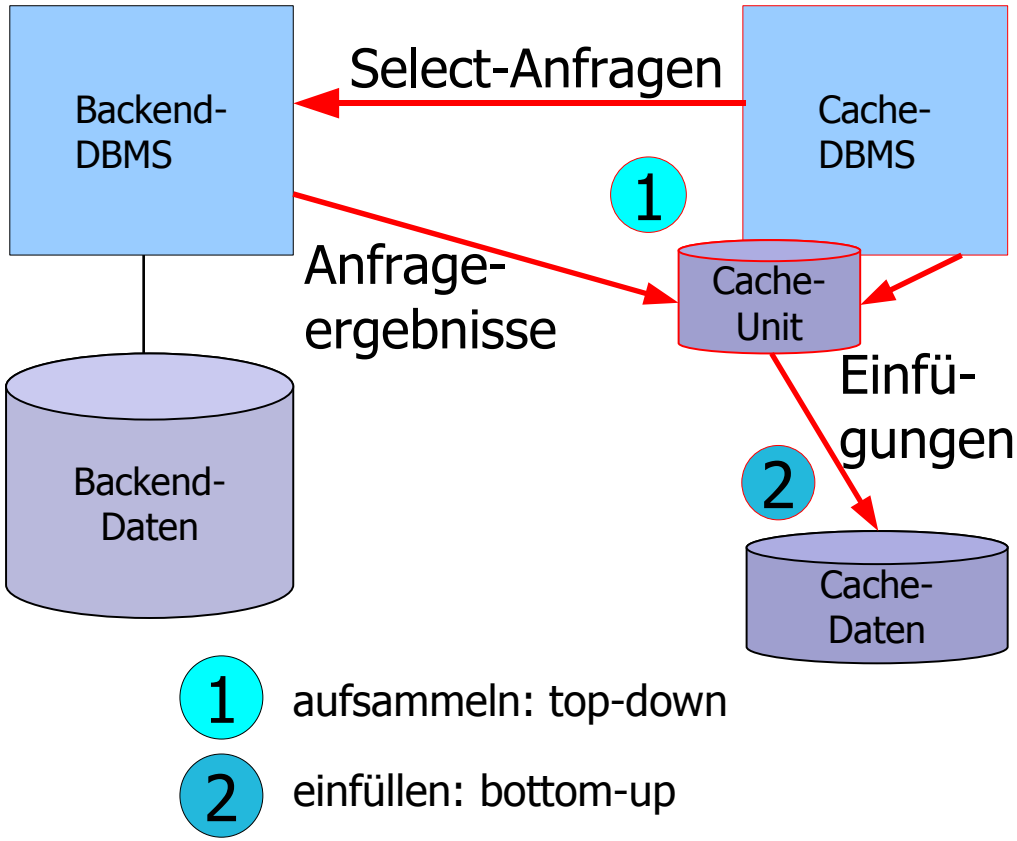
Problem: Verteilung

Indirektes Laden



Problem: Verteilung

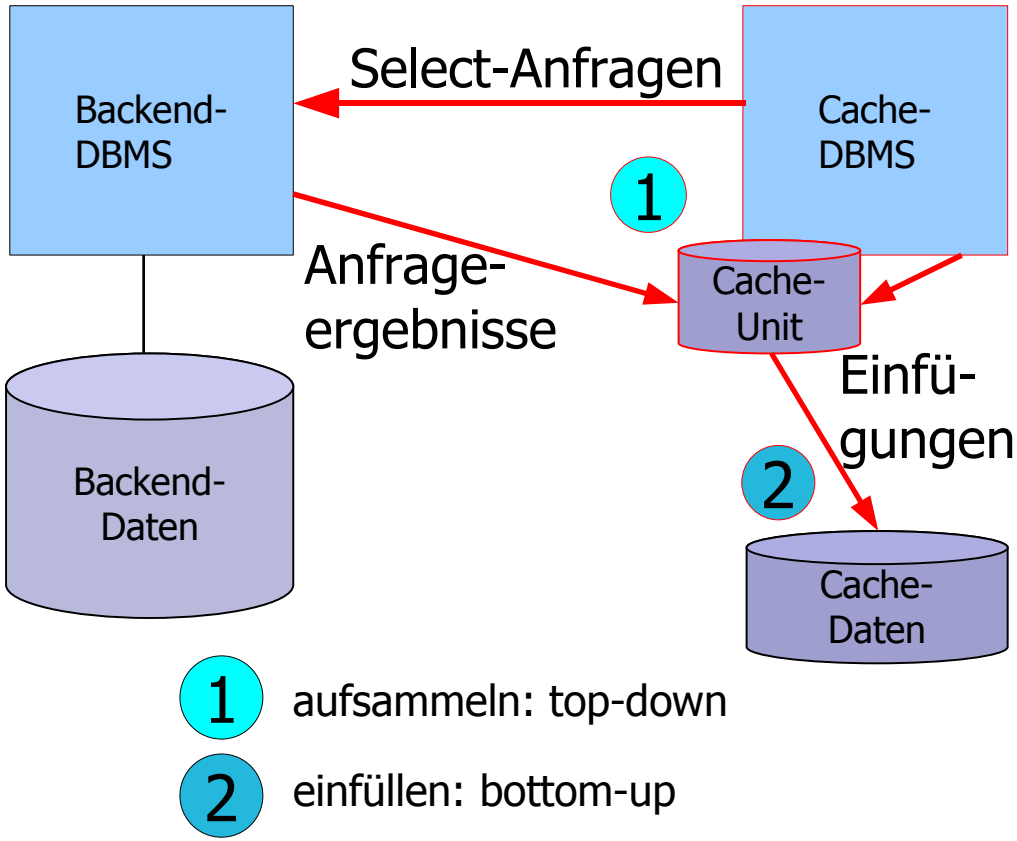
Indirektes Laden



✗ Indirektes Einladen

Problem: Verteilung

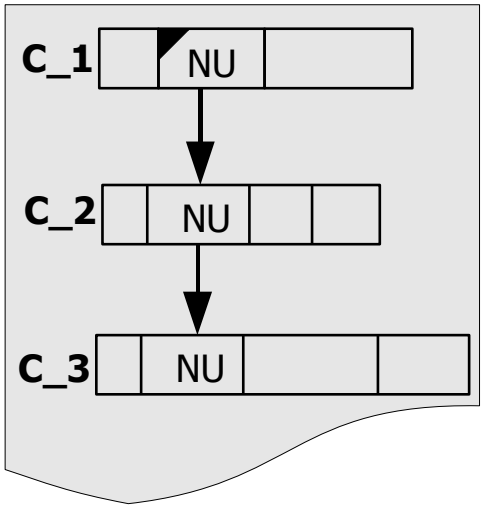
Indirektes Laden

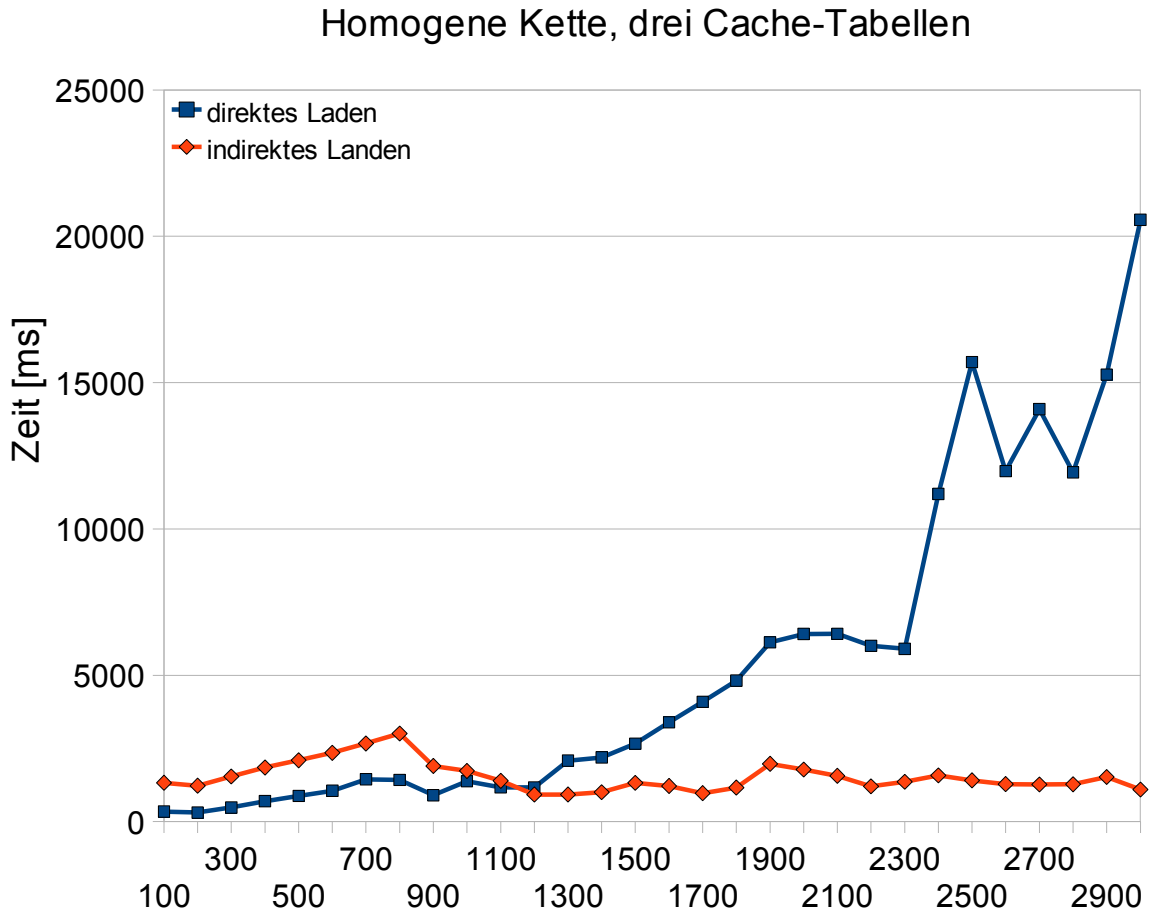


- ✗ Indirektes Einladen
- ✗ Mehrfaches Anfragen vom Cache an Backend
  ➔ Latenzproblematik

Ergebnisse

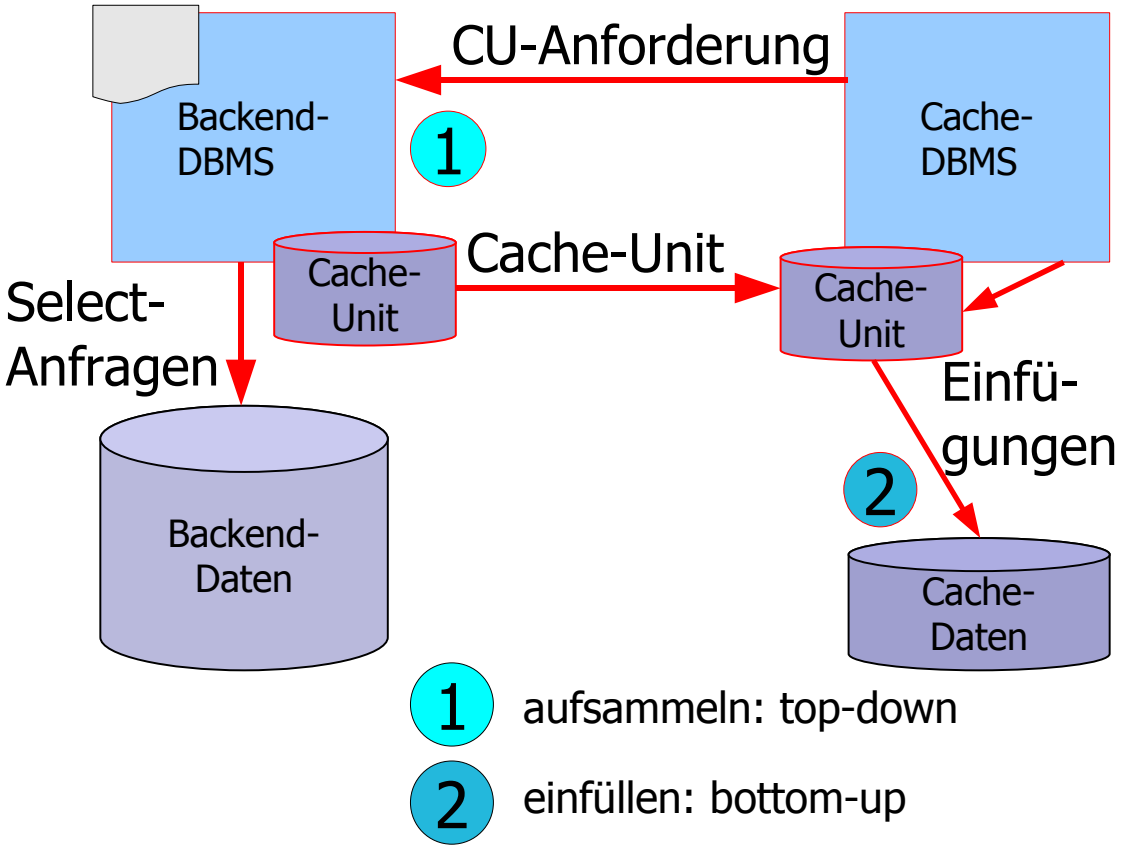
Direktes Laden vs. indirektes Laden





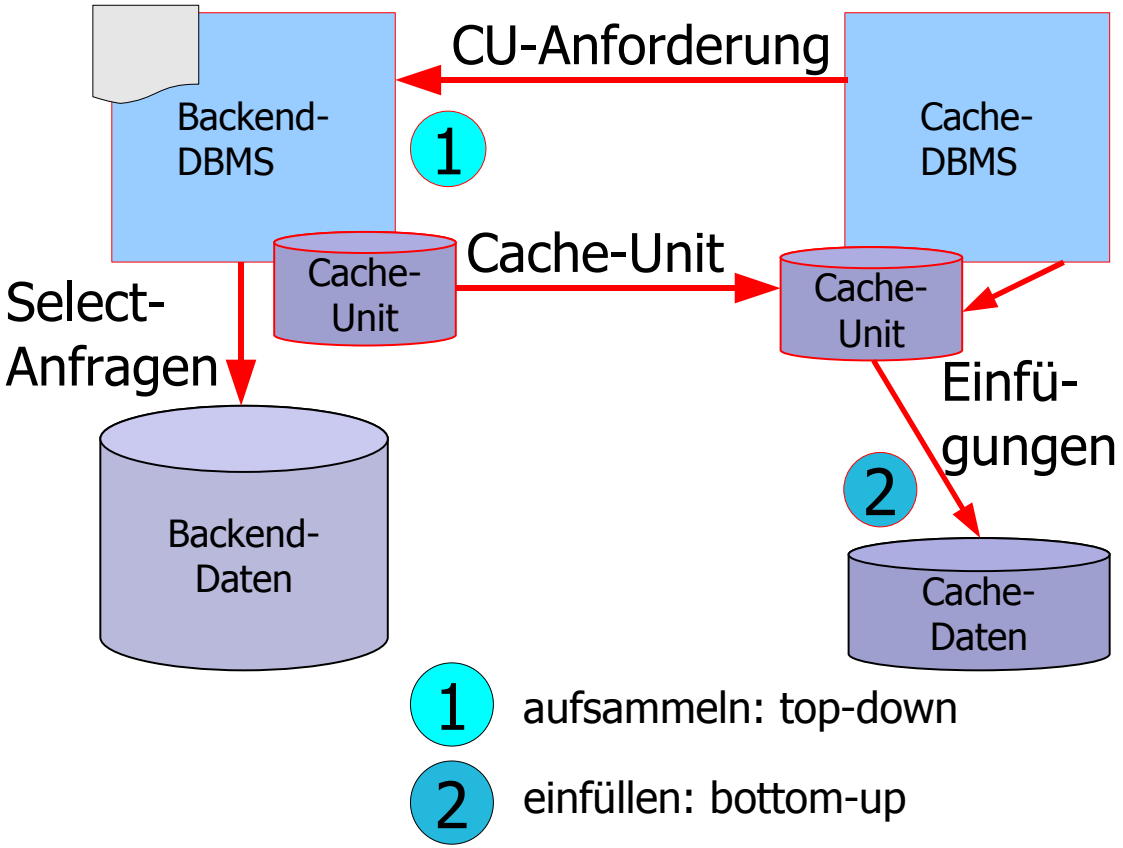
Problem: Verteilung

Vorbereitetes, indirektes Laden



Problem: Verteilung

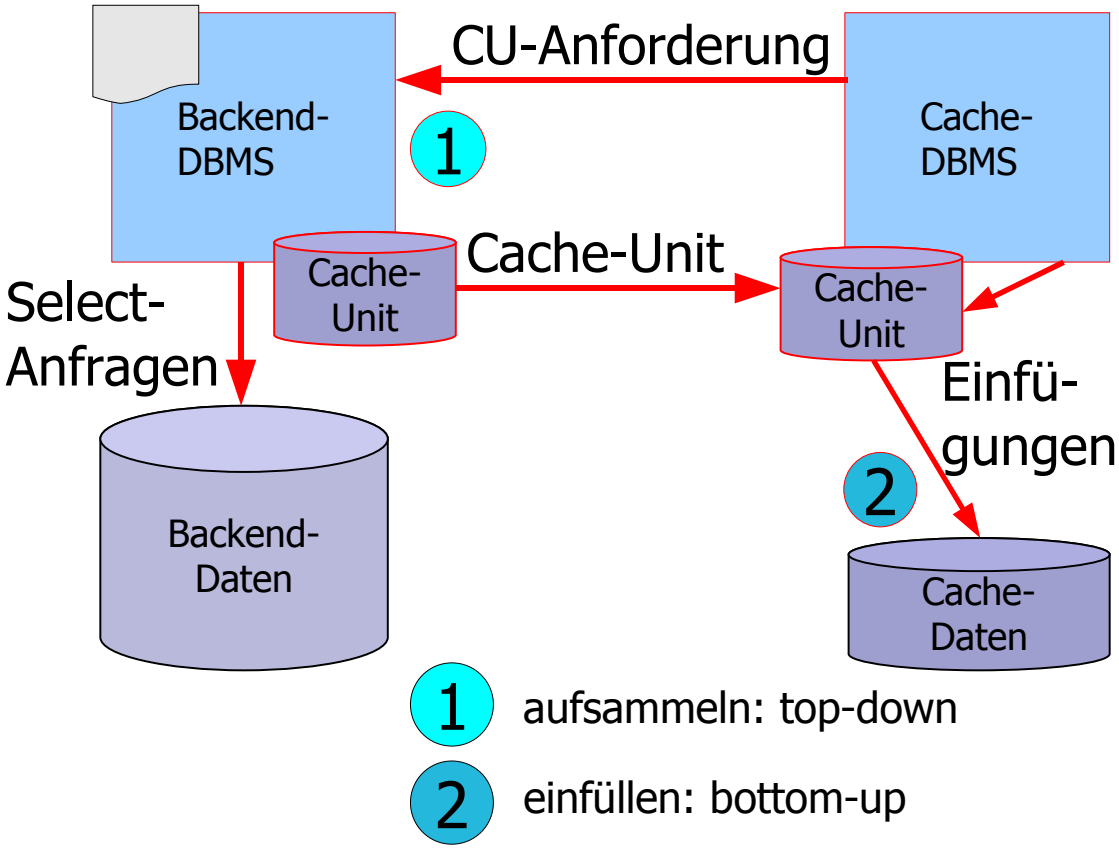
Vorbereitetes, indirektes Laden



- ✔ Konstante Prädikatlänge in Anfragen

Problem: Verteilung

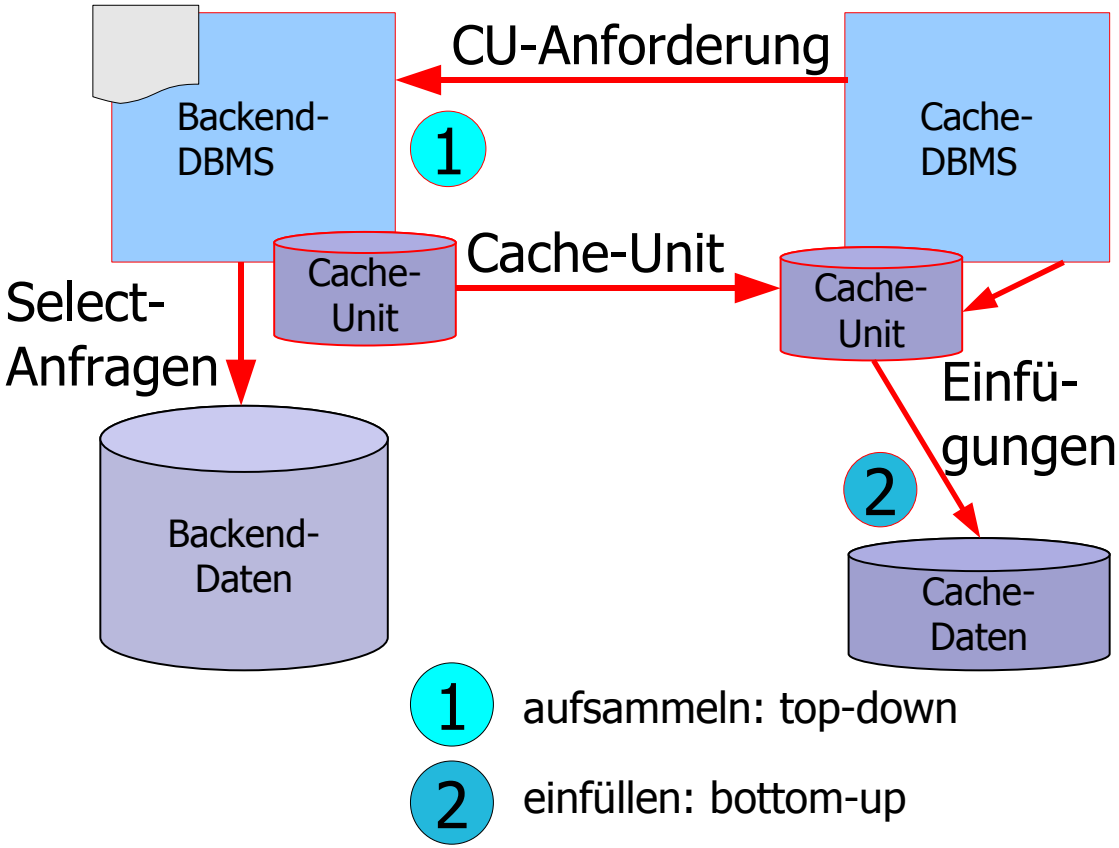
Vorbereitetes, indirektes Laden



- ✔ Konstante Prädikatlänge in Anfragen
- ✔ Geringe Abhängigkeit vom Füllgrad der Cache-Tabellen

Problem: Verteilung

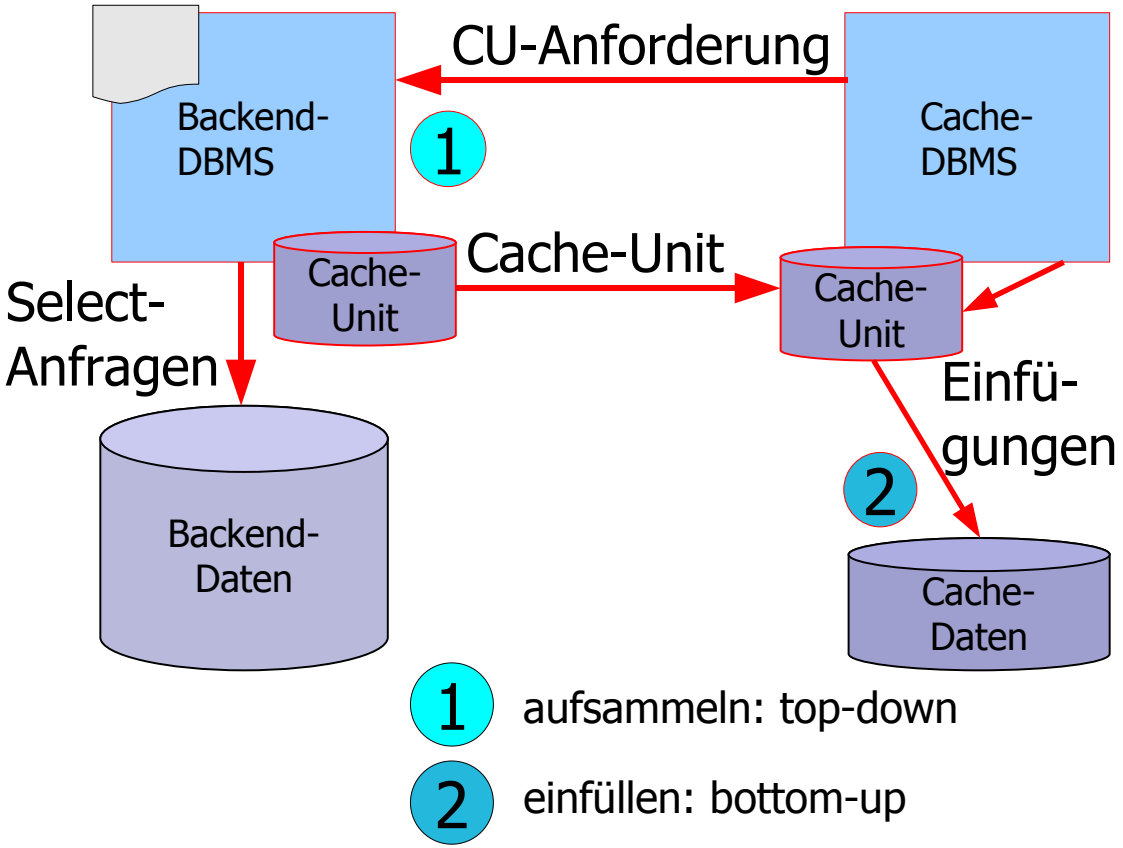
Vorbereitetes, indirektes Laden



- ✔ Konstante Prädikatlänge in Anfragen
- ✔ Geringe Abhängigkeit vom Füllgrad der Cache-Tabellen
- ✔ Einmalige Anfrage an das Backend

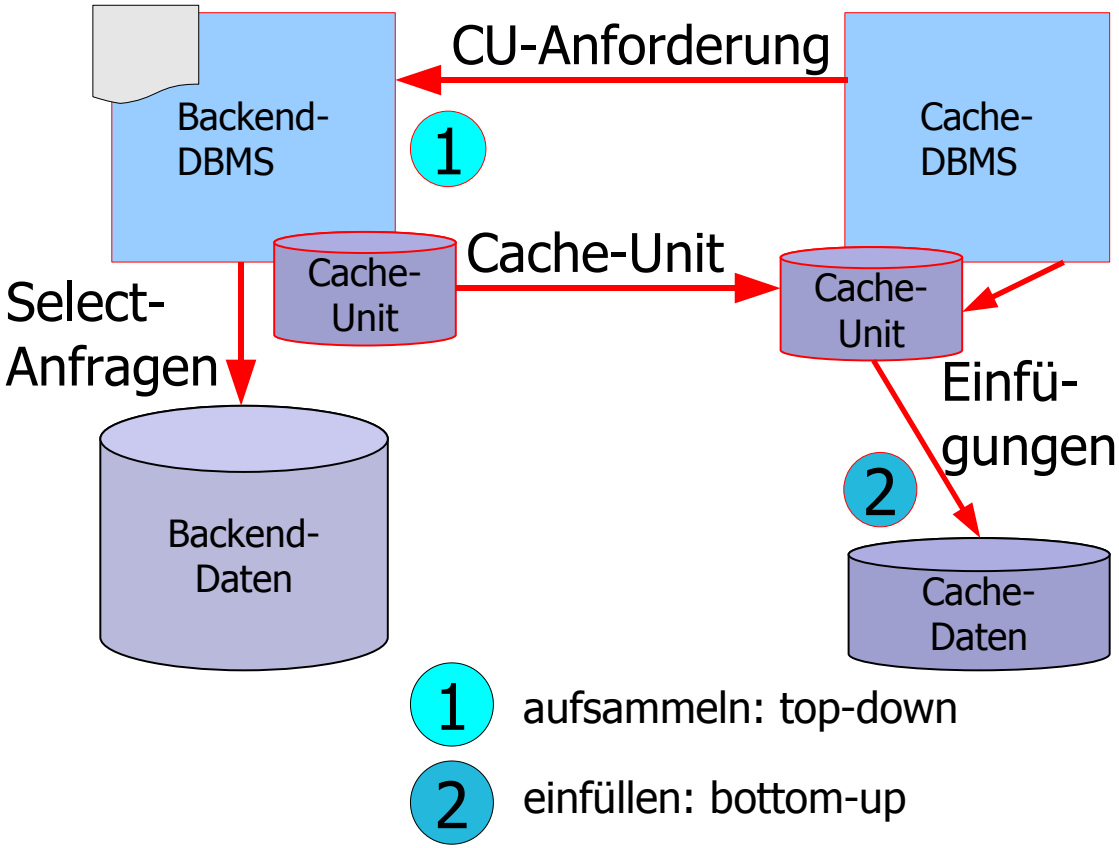
Problem: Verteilung

Vorbereitetes, indirektes Laden



Problem: Verteilung

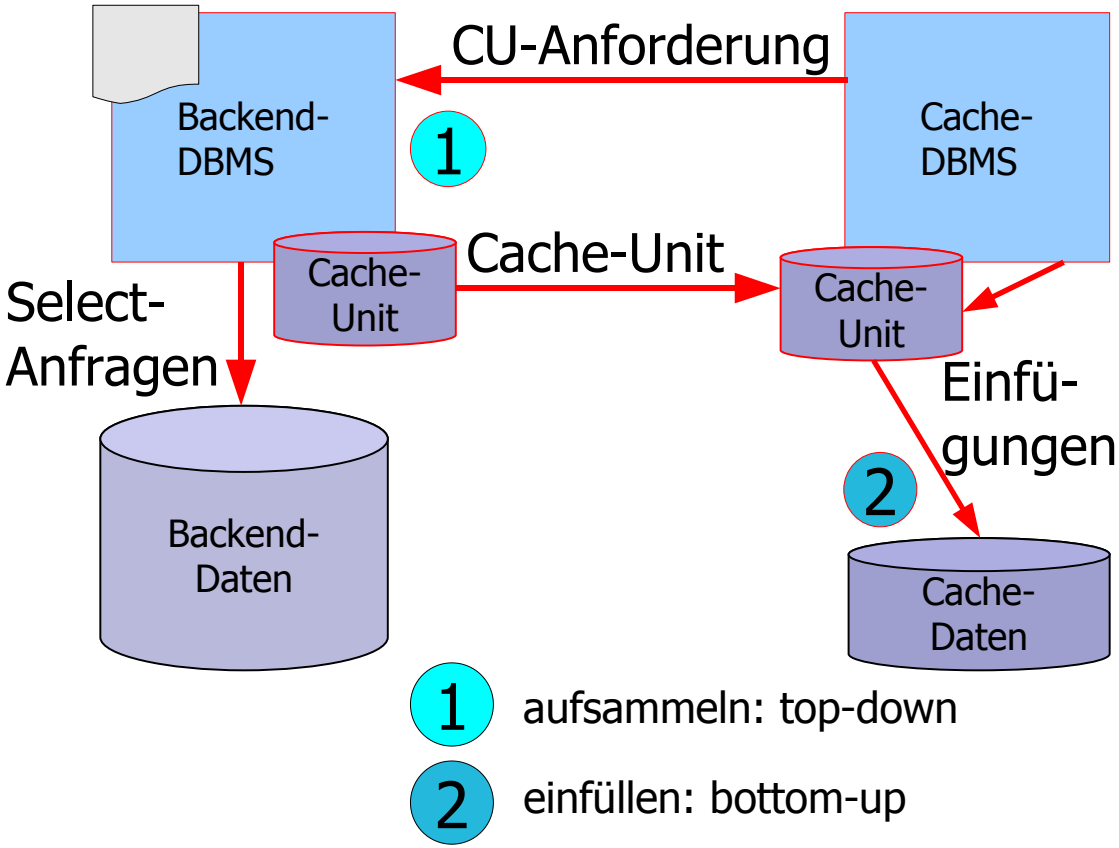
Vorbereitetes, indirektes Laden



✗ Indirektes Einladen

Problem: Verteilung

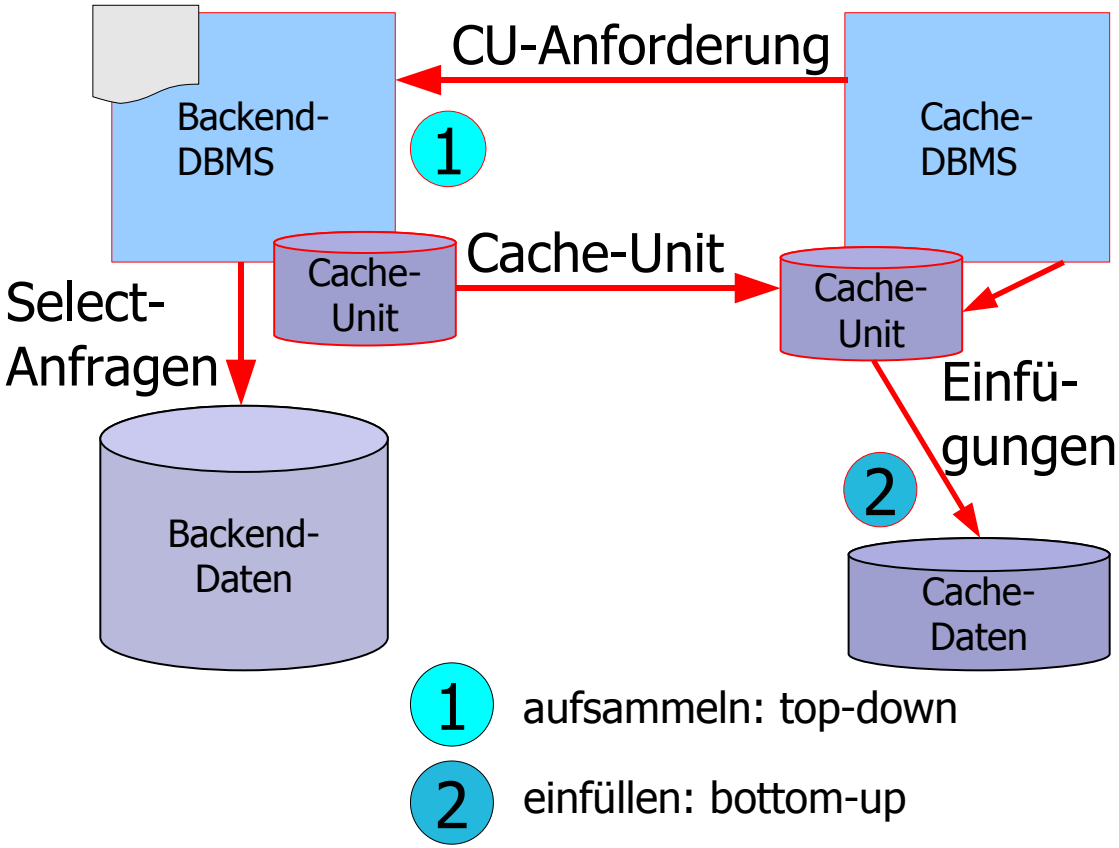
Vorbereitetes, indirektes Laden



- ✗ Indirektes Einladen
- ✗ Backend muss Cache-Group-Definitionen kennen

Problem: Verteilung

Vorbereitetes, indirektes Laden

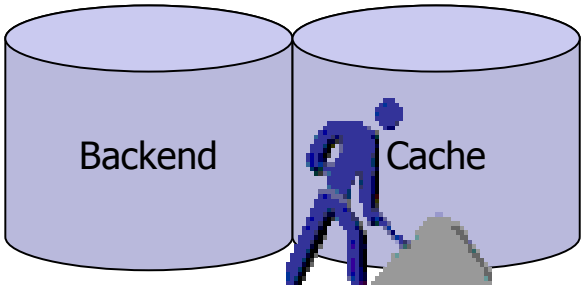


- ✗ Indirektes Einladen
- ✗ Backend muss Cache-Group-Definitionen kennen
- ✗ Backend muss das Aufsammeln der Daten organisieren

Fazit: Verteilung

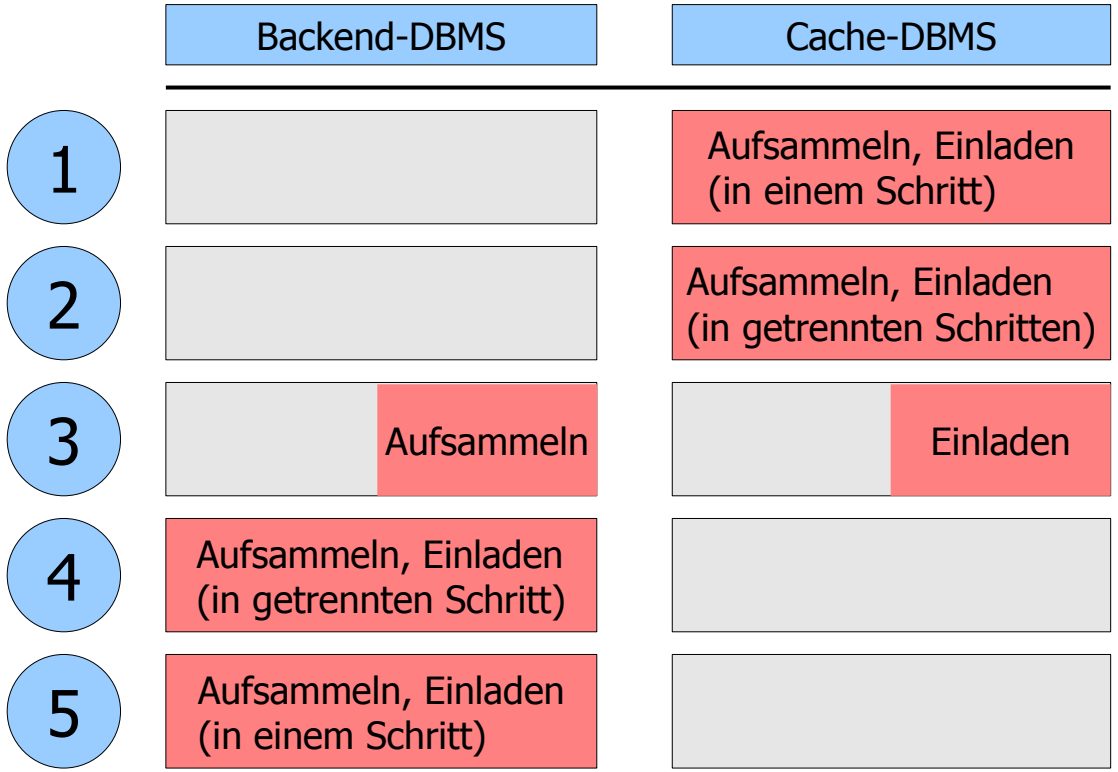
- Zentral:
  - Aufwand für die Verwaltung der verteilten Komponenten?
  - Verwaltung präziserer Metadaten:
    - bietet gezieltere Verarbeitungsmöglichkeiten
    - verursacht größeren Verwaltungsaufwand
- noch wesentlich komplizierter:
  - Synchronisation, Adaption

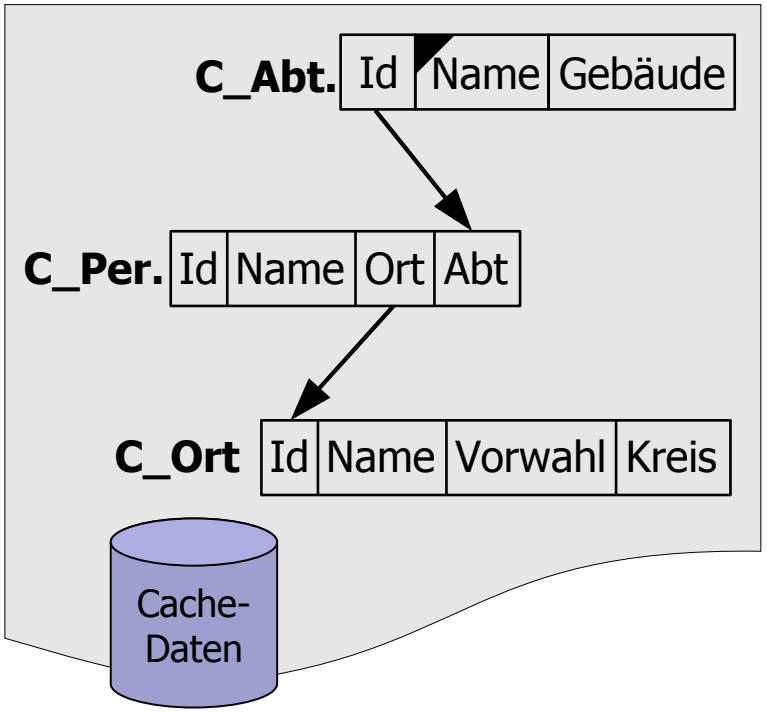
Den Müll – bringt der Cache raus!



Problem: Verteilung

Beispiel: Ladevorgang (1)





Problem: Verteilung

Vor- und Nachteile

	1	2	3	4	5
Vorteile	- Backend ist entlastet - direktes Einladen - Backend benötigt keinerlei zusätzliche Metadaten	- Backend ist entlastet - Backend benötigt keinerlei zusätzliche Metadaten - kleine Abhängigkeit vom Füllgrad der Cache-Tabellen	- einmalige Anfrage - Backend verwaltet Metadaten/Wissen - kleine Abhängigkeit vom Füllgrad der Cache-Tabellen	- kleine Abhängigkeit vom Füllgrad der Cache-Tabellen - Cache ist entlastet	- Cache ist entlastet - direktes Einladen - große Abhängigkeit vom Füllgrad der Cache-Tabellen
Nachteile	- unkontrollierbar lange Prädikate in Anfragen - mehrfache Anfragen - große Abhängigkeit vom Füllgrad der Cache-Tabellen - Backend hat kein Wissen (z. B. über den Füllzustand)	- indirektes Einladen - mehrfache Anfragen - Backend hat kein Wissen (z. B. über den Füllzustand)	- indirektes Einladen - Backend teilweise belastet	- indirektes Einladen - mehrfache Einfüge-operationen - Backend benötigt komplettes Wissen - Backend ist stark belastet	- unkontrollierbar lange Prädikate in Anfragen - mehrfache Einfüge-operationen - Backend benötigt komplettes Wissen - Backend ist stark belastet