

Grundlagen der Datenbanktechnik

Martin Klier, DBA
martin.klier@unix.net

Referent

- Martin Klier, 28
- Linux- und Datenbankadministrator
- Schwerpunkt hochverfügbare Systeme, Cluster und Replikation
- Arbeitgeber: A.T.U Weiden
- Linux seit 1997
- Oracle Database seit 2003
- etwas IBM DB2
- Kontakt: martin.klier@unix.net
- Web: <http://www.usn-it.de>



Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Do you speak English?

IT generell und Datenbanken im Speziellen wurden in den USA „erfunden“

Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Was ist eine Datenbank?

„Die wesentliche Aufgabe eines Datenbanksystems ist es, große Datenmengen effizient, widerspruchsfrei und dauerhaft zu speichern und benötigte Teilmengen in unterschiedlichen, bedarfsgerechten Darstellungsformen für Benutzer und Anwendungsprogramme bereitzustellen.“ (Wikipedia)

 **Strukturierte Datenablage**

Geschichte I

- 1970 Theoretische Grundlagen von Edgar F. Codd (IBM Almaden)
- 1974 Ingres RDBMS (QUEL)
- 1977 IBM System R, Sonderanfertigung für Pratt&Whitney (SEQUEL)
- 1977 SDL (Vorläufer von Oracle) erstellt SEQUEL-RDMBS für C.I.A.
- 1979 RSI (ex SDL) bringt „Oracle RDBMS 2.0“ auf den Markt (SQL)
- 1984 Erstes RDMBS mit Lesekonsistenz

Geschichte II

- 1987 RDBM-Systeme für UNIX kommen auf den Markt
- 1988 Prozedurale Programmierung möglich (PL/SQL)
- 1995 64bit-RDBMS erscheinen
- 1997 Datenbanken werden für Internetapplikationen optimiert
- 1999 XML-Unterstützung
- 2003 Datenbank-Server-Cluster mit voller Cachekohärenz
- 2005 Ableger der großen Enterprise-RDBMS kostenlos verfügbar

Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Praxisbeispiel

→ ER-Diagramm

→ SELECT:

```
select a.KD_NR, a.NAME, a.VORNAME  
from TAB_KUNDEN a  
where a.NAME='Schneider';
```

Praxisbeispiel

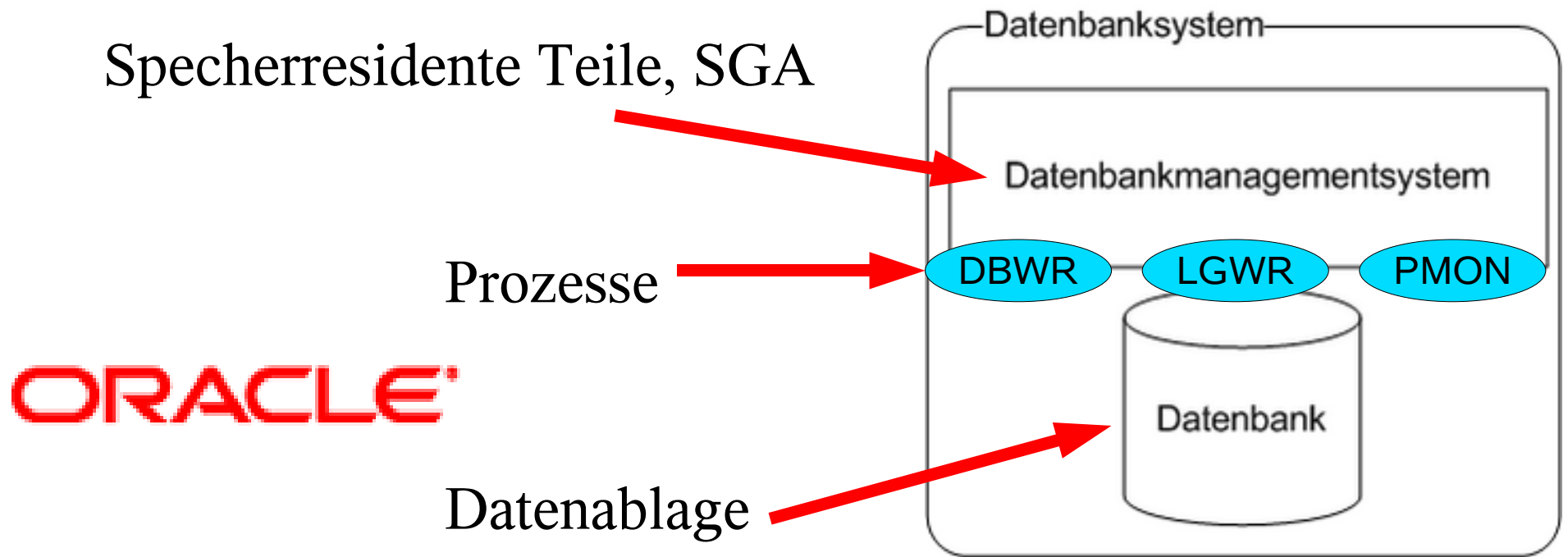
→ SELECT + Join:

```
select b.AUFTRAG_NR  
from TAB_KUNDEN a, TAB_AUFTRAEGE b  
where a.KD_NR=b.KD_NR;
```

→ UPDATE:

```
update TAB_KUNDEN a  
set a.NAME='Schneider' where a.NAME = 'Shnyder';
```

Aufbau DB-System



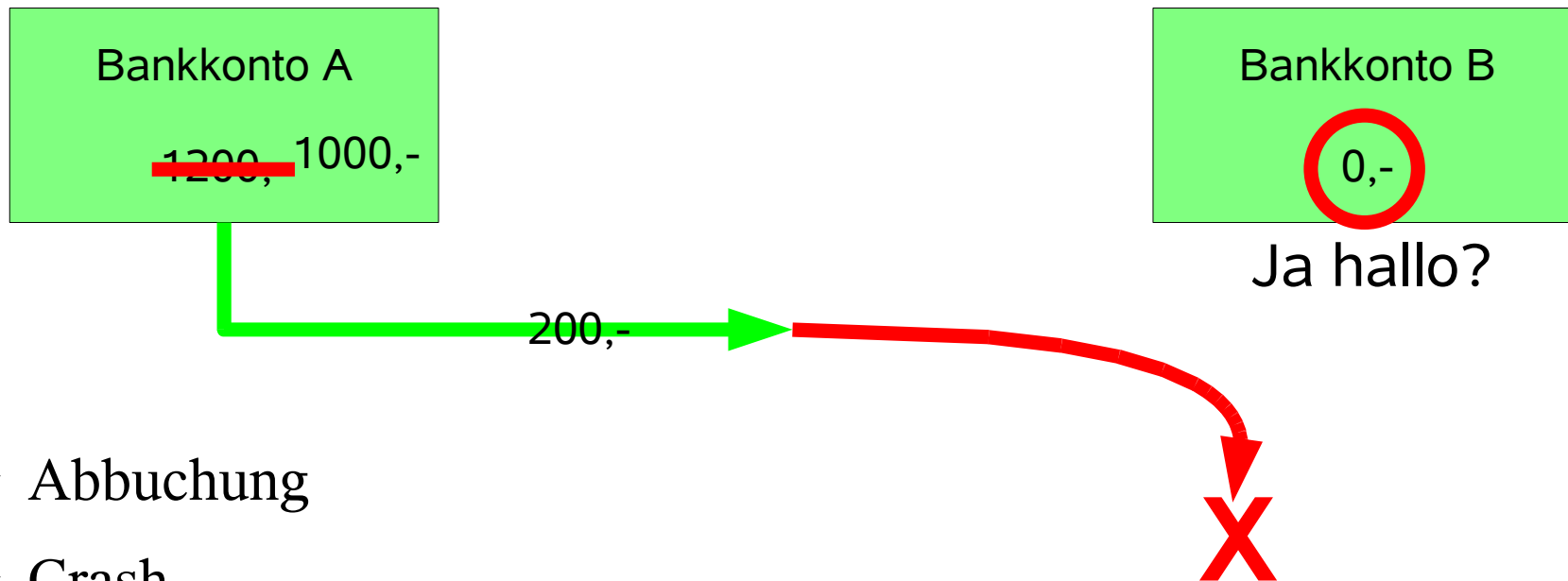
Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Transaktionsprinzip

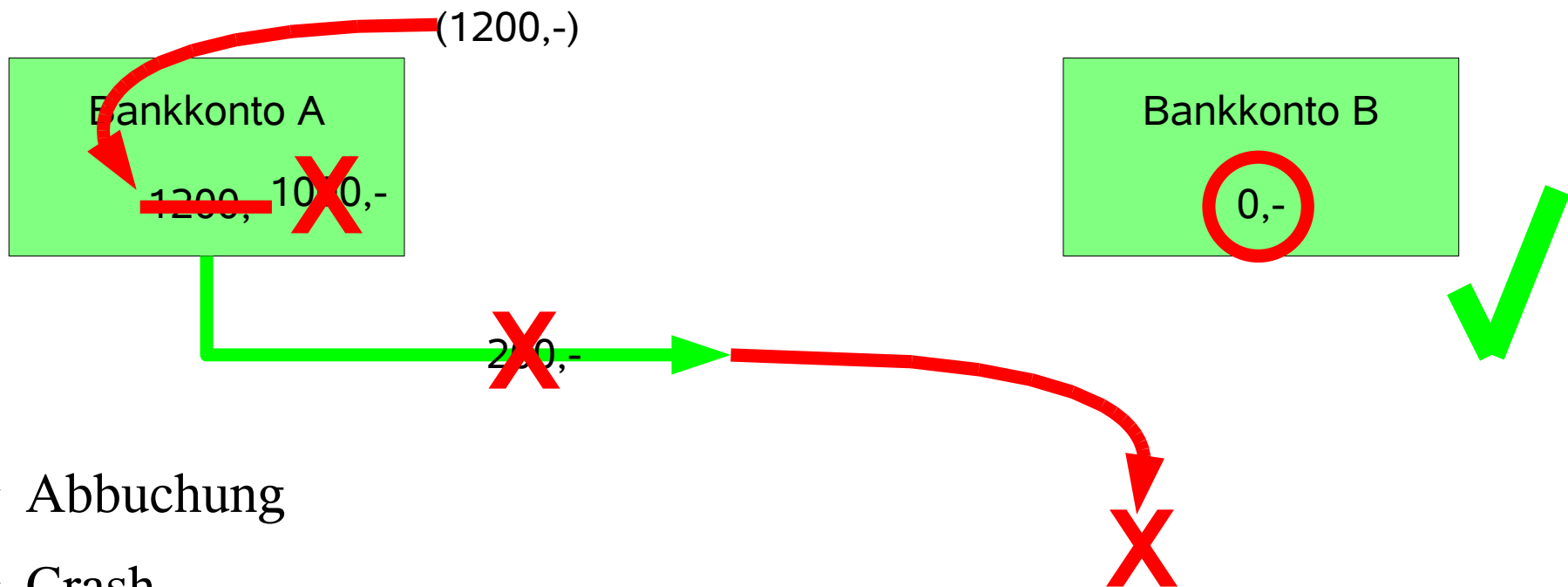
„Eine Transaktion fasst in sich mehrere logisch zusammengehörige, jedoch nicht zwangsweise technisch voneinander abhängige Arbeitsschritte zusammen“

Transaktion - Beispiel



- Abbuchung
- Crash
- Was nun?


Absicherung



- Abbuchung
- Crash
- Zurück!

Commit / Rollback

```
update KONTEN where KONTO_NR=A set SALDO=1000.00;  
update KONTEN where KONTO_NR=B set SALDO=200.00;  
commit;
```



```
update KONTEN where KONTO_NR=A set SALDO=1000.00;  
update KONTEN where KONTO_NR=B set SALDO=200.00;  
##### update ERROR 890 - USER NOT SANE #####  
rollback;
```

Ablauf: Abfrage

SELECT

- Auswerten des SQL-Befehls
- Ermitteln der Menge betroffener Daten
- Einlesen der Daten in den Cache
- Formulierung der Ausgabe
- Weiterleitung an den User

Ablauf: Update

UPDATE + COMMIT

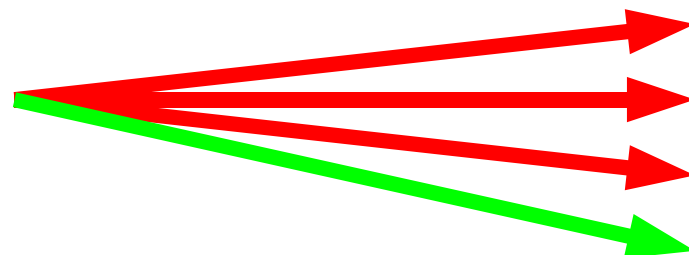
- Auswerten des SQL-Befehls
- Ermitteln der Menge betroffener Daten
- Einlesen der Daten in den Cache
- Erstellen von Sicherheitskopien
- verändern der betroffenen Blöcke
- Freigeben der geänderten Daten (für andere Lesezugriffe)

Theorie zum Join

*select b.AUFTRAG_NR from TAB_KUNDEN a, TAB_AUFTRAEGE b
where a.KD_NR=20 and a.KD_NR=b.KD_NR;*

TAB_KUNDEN

KD_NR	NAME
100	Schneider
20	Malzer
...	
13	Mayer



TAB_AUFTRAEGE

A_NR	KD_NR	BEZ
5233	97	Projekt
5253	13	Service
....		
5243	20	Ersatz

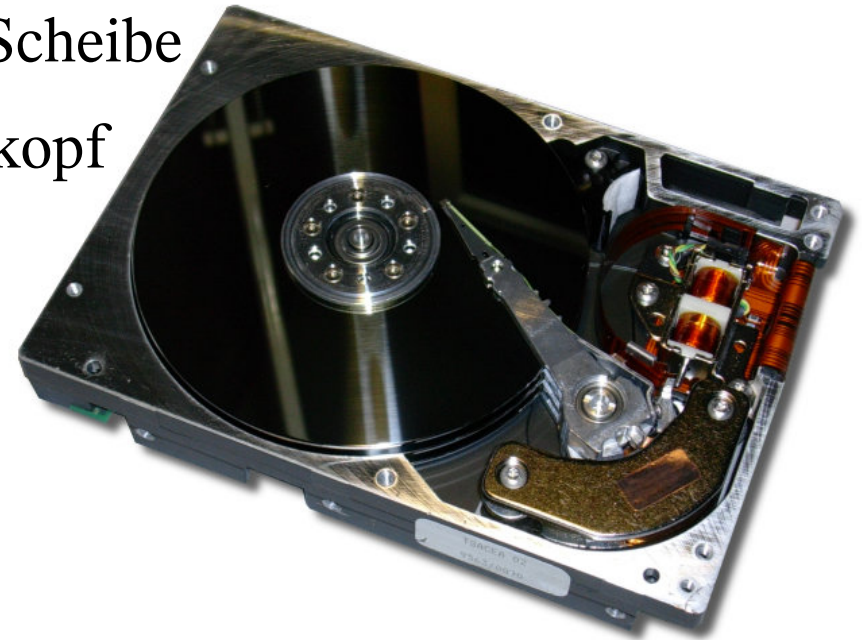
Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

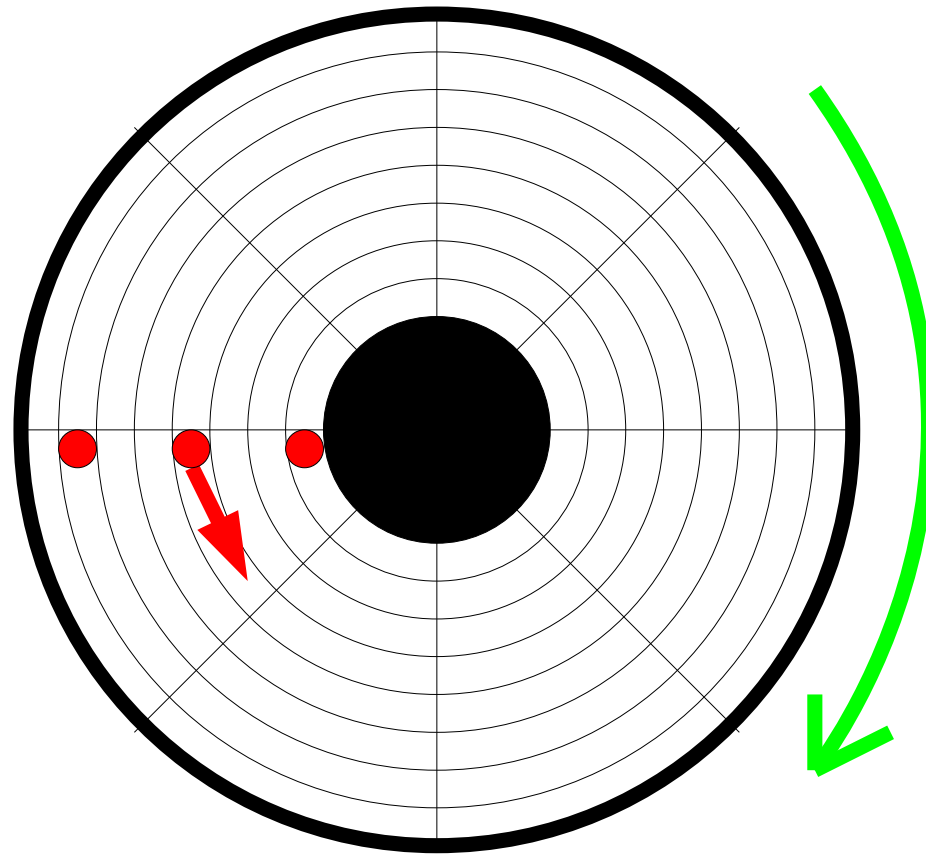
Exkurs: Medien

Funktionsweise einer Festplatte

- Magnetisierbare, sehr schnell drehende Scheibe
- Mechanisch „gefährlicher“ Schreib-/Lesekopf
- Schnell? Langsam?
- Welche Einflußgrößen?



Exkurs: Medien



Exkurs: Medien

Größenordnungen zu RAM und Festplatten

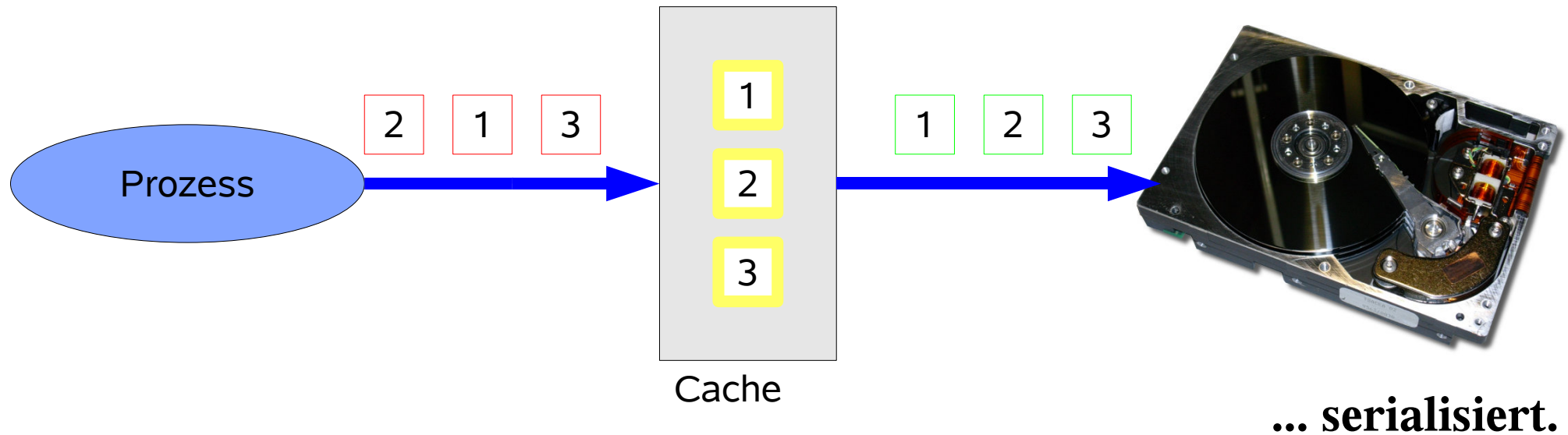
- Handelsübliche Größen
- Preis pro Gigabyte (GB)
- mittlere Zugriffszeit auf Daten

Exkurs: Medien

	Größe	Preis pro GB	mittlere Zugriffszeit
RAM-Modul	2GB - 8GB	20 EUR - 150 EUR	0,000 000 1s (100ns)
Server-RAM	16GB - 256GB	dto.	dto.
Festplatte	146GB - 1TB	0,32 EUR - 0,24 EUR	0,008s - 0,012s (~10ms)
RAID-Array Kapazität	146GB – 10TB	1,44 EUR - 10 EUR (!!)	0,001s - 0,010s (~8ms)

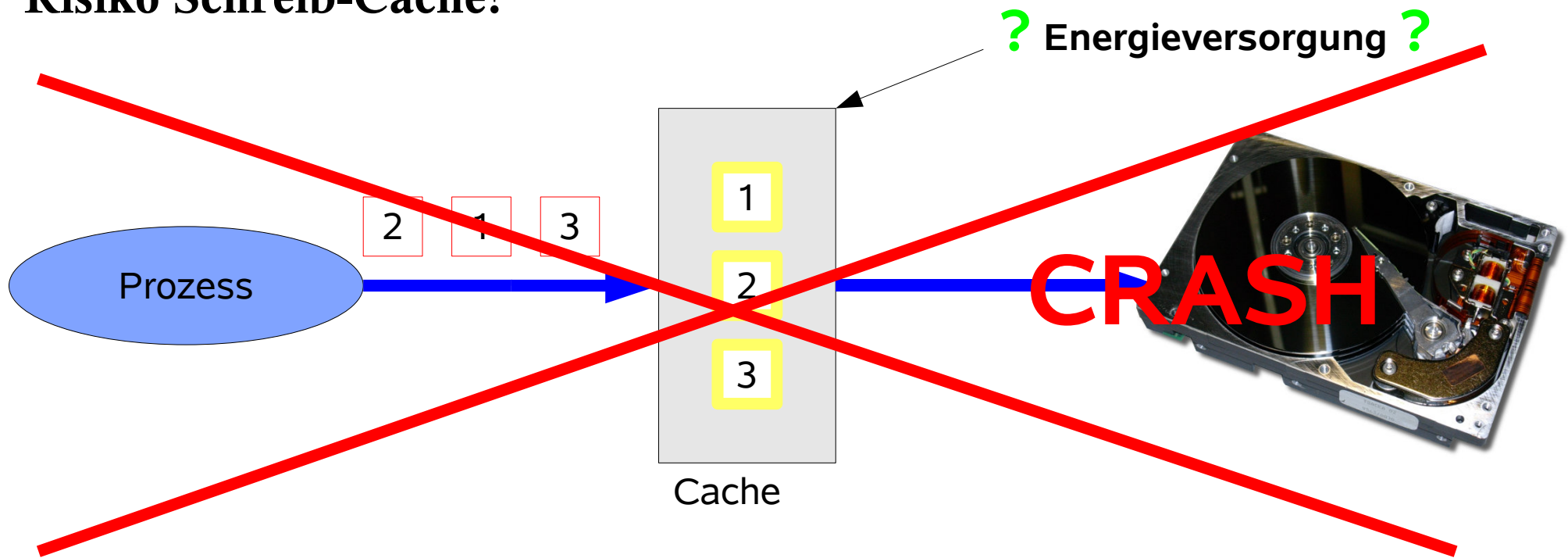
Write-Caching

Schreib-Cache (RAM vor Disk) ...



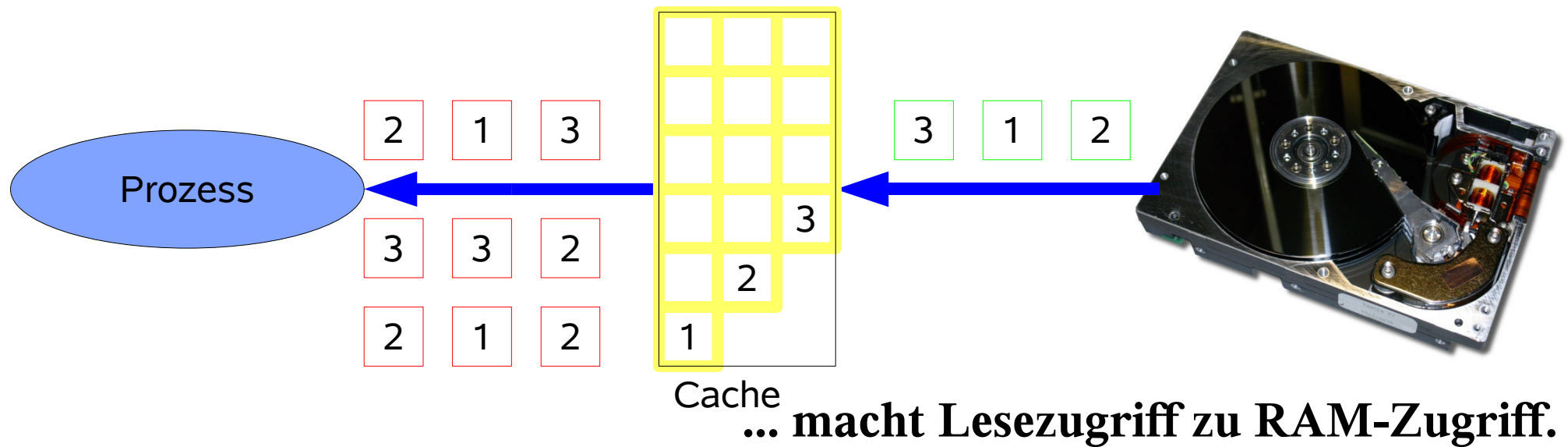
Write-Caching

Risiko Schreib-Cache!



Read-Caching

Lese-Cache (RAM nach Disk) ...

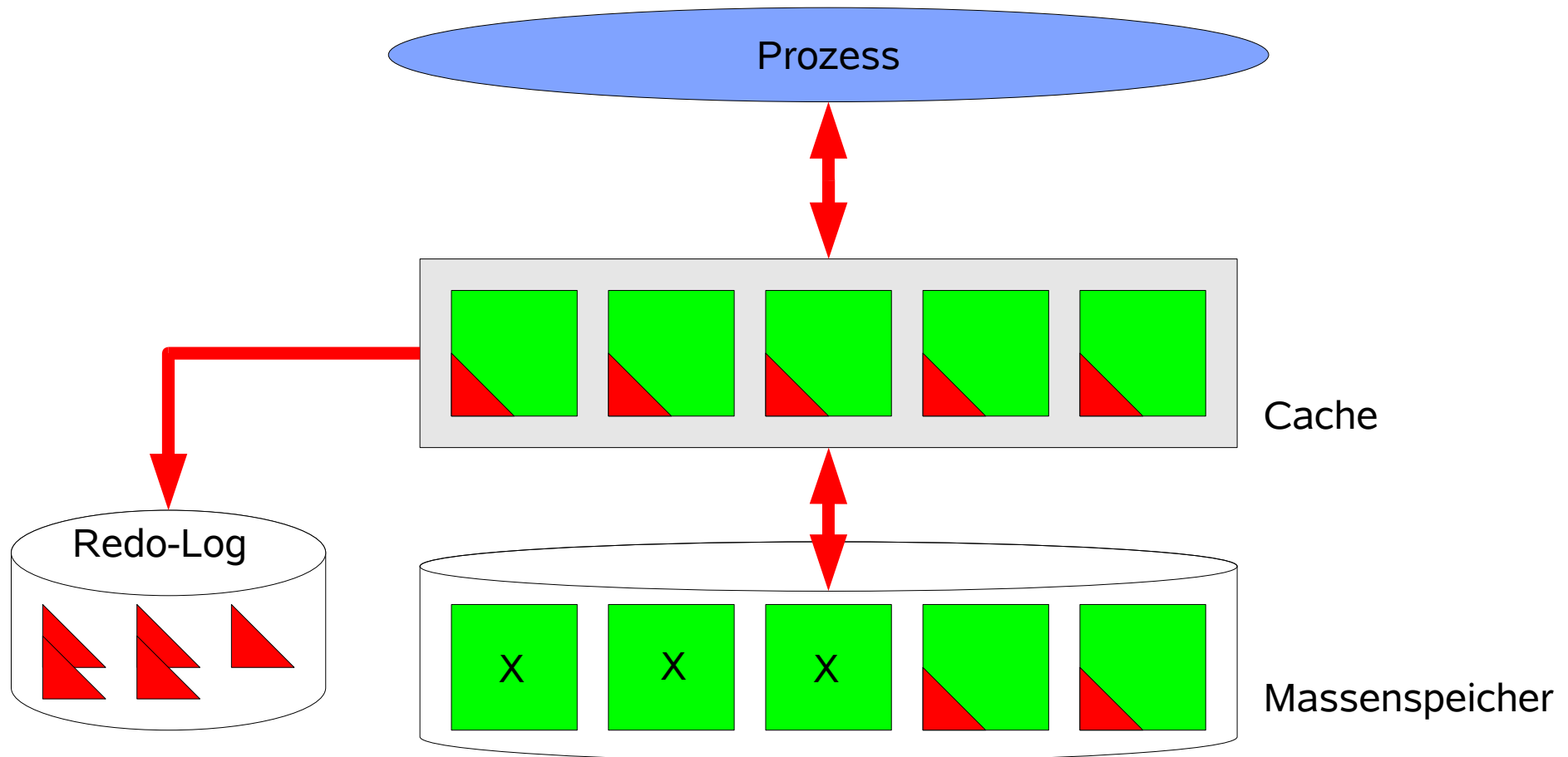


Cache absichern

Datenbankcache im Arbeitsspeicher des Servers

- kein Backup des RAM, d.h. Totalverlust bei Crash + Neustart
- daher Kompromiss notwendig
- Schreiben von „Delta“-Werten auf schnelle Festplatten

Cache absichern



Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Klassifikation I

→ **Tabellenkalkulation**

→ Excel, OpenOffice Calc

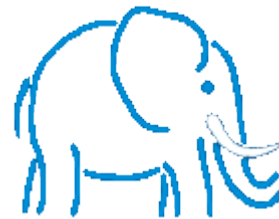
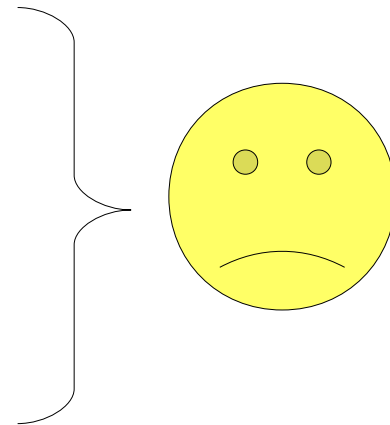
→ **Workstation-Level**

→ Access, BerkeleyDB

→ **Mittelklasse**

→ MySQL

→ PostgreSQL



Klassifikation II

Enterprise-Niveau

- Oracle
- IBM DB2
- Microsoft SQL-Server
- IBM Informix
- Sybase
- SAP-DB

ORACLE®



Microsoft®

Informix®

SYBASE®



„Global Player“

Überblick - Klassen

Produktklasse	Vorteile	Nachteile
Tabellenkalkulation	schnell erlernbar wenig Aufwand	kein SQL Strukturschwächen
Workstation-Niveau	geringe Größe wenig Administration	keine Mehrbenutzerfähigkeiten fehlende Absicherung (Cache usw.)
Mittelklasse	kostenlos viele Fähigkeiten	schwache Transaktionssicherheit Sperr- bzw. Konsistenzprobleme
Enterprise-Level	tausende Features exzellenter Support	sehr teuer (~30.000 EUR / CPU) Admin: lange Einarbeitungszeit

Agenda

- 1) Überblick
- 2) Einführung
- 3) Technische Realisierung
- 4) Transaktionen
- 5) Cache
- 6) Klassifikation Datenbanksysteme
- 7) Anwendung in der Wirtschaft

Anwendung - OLTP

Online Transaction Processing – Echtzeitbetrieb, viele kleine Transakt.

- Kunden-, Adress- und Artikeldatenbank, (Online-)Shop
- Website-backend (dynamische Webseiten)
- Finanzbuchhaltung
- Automatisiertes Lager / Logistik / Warenwirtschaft

Response time, response time, response time!

Anwendung - OLAP

Online Analytical Processing – Data Warehouse, wenige große Transakt.

- Auswertungen zur Wirtschaftlichkeit, Controlling
- Auswertungen zur Güterverteilung
- Kundenprofil-Erstellung
- u.U. Rasterfahndung (BKA/LKA)

„Erkaufen von Zeit - durch Aufwand von Massenspeicherplatz“

Danke

... für Ihre Aufmerksamkeit!

Noch Fragen offen? Dann los!

Schönes Wochenende!