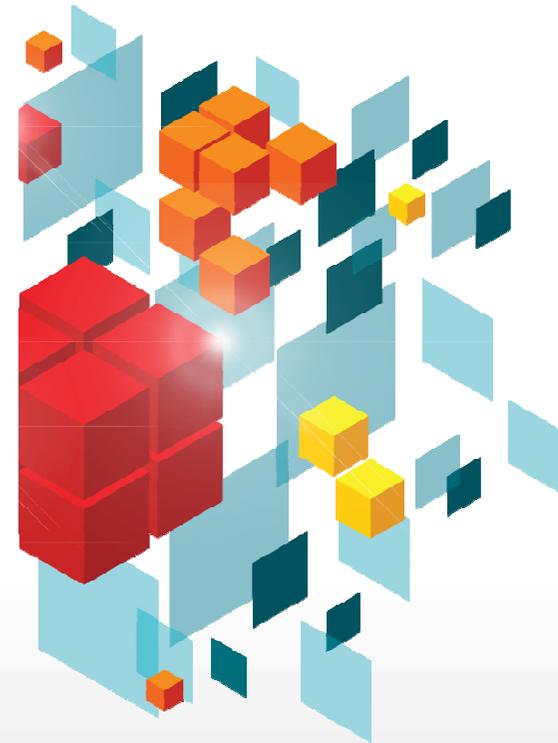


## **IBM Performance 2011**

**Minimal-invasives Werkzeug zur extremen  
Beschleunigung von Datenbankabfragen**

**Prof. Dr. Wolfgang Benn**





# Agenda

- Das Werkzeug: ***dimensio***
- Funktionsweise
- Performancegewinn
- Minimal-invasive System-Integration



# *dimensio*





## Was ist *dimensio*

- *dimensio* ist ein mehrdimensionaler Datenbankindex
  - Dimension = Attribut
  - Mehrdimensional = SQL-Query mit langer Prädikatliste
  - Lange Prädikatliste = Korrelation vieler Attributwerte
  
- *dimensio* ist ein semantischer Datenbankindex
  - semantisch = Intervallbildung mit Bezug auf den DB-Inhalt
    - Datensätze mit ähnlichem Inhalt bilden eine Gruppe (Cluster)
    - Cluster bilden die Intervallgrenzen
  
- *dimensio* ist ein Produkt der *dimensio informatics GmbH*

## Funktionsweise



- *dimensio* lernt die Zusammenhänge der Datensätze
  - Lernvorgang analog zum Menschen
    - unüberwacht, d.h. es gibt keine Vorgaben, wie zu lernen ist
    - ausschließlich abhängig von dem, was vorgefunden/angeboten wird
  - Technologie aus der Künstlichen Intelligenz
    - Künstliches Neuronales Netz
    - Weiterentwicklung von Kohonens SOMs (1982)
    - Projektion eines hochdimensionalen auf einen niederdimensionalen Raum
  
- *dimensio* verwaltet das Gelernte in einer effizienten DS
  - Baumartige Datenstruktur analog zum R-Baum



## Was ist *dimensio* nicht

- Die eierlegende Wollmilchsau
- Probleme gibt es bei
  - Geringer Selektivität der Anfragen
    - Hier versagen allerdings alle Indexe
  - Schlechter Modellierung des Anwendungsgebietes
    - Ein Problem der KI gemeinhin und der KNN insbesondere
  - Unvollständiger Auswahl der Dimensionen
    - Man sollte wissen, was man will...

aber:

- *dimensio* wird gerade zertifiziert als





# Funktionsweise



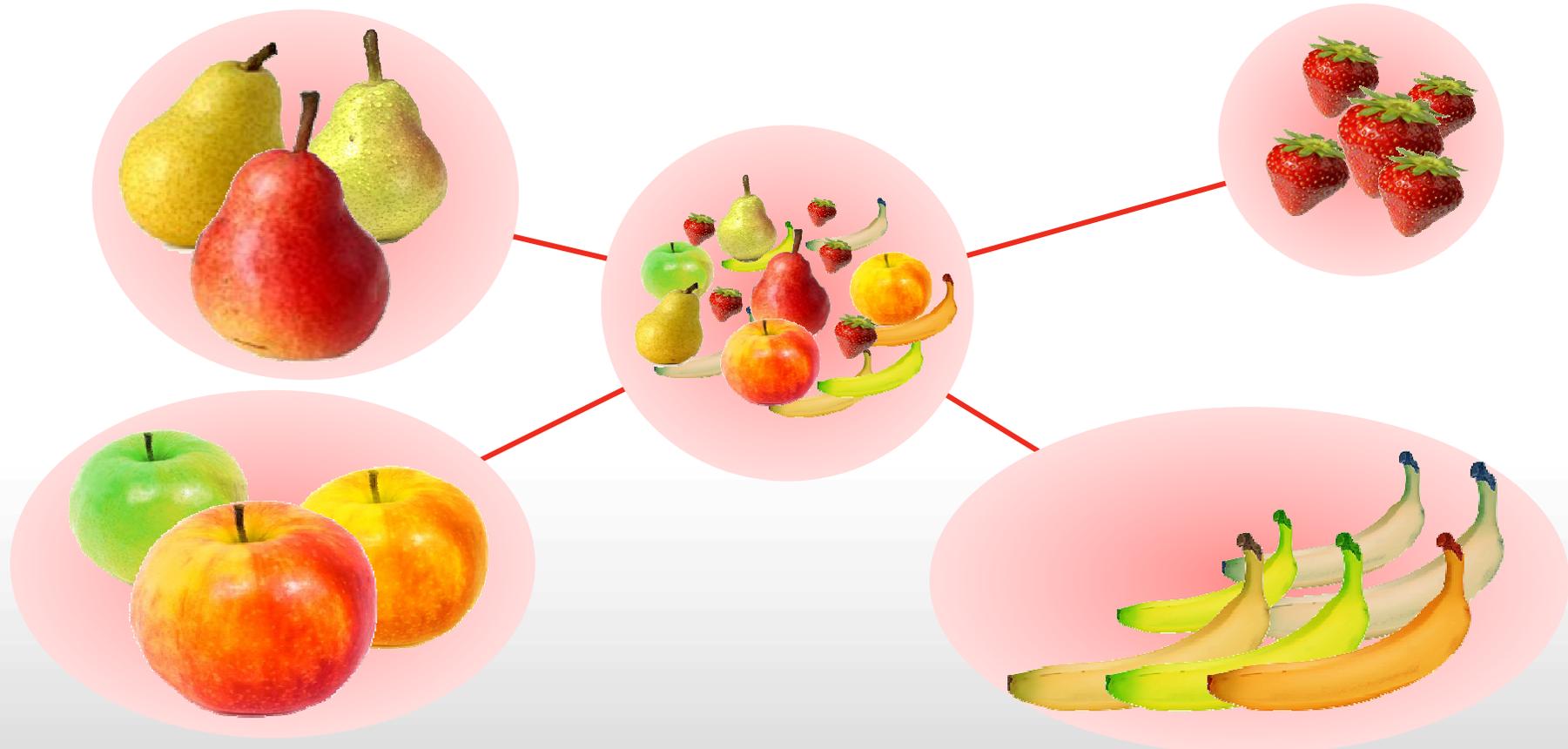
# Beispiel-Datenbank



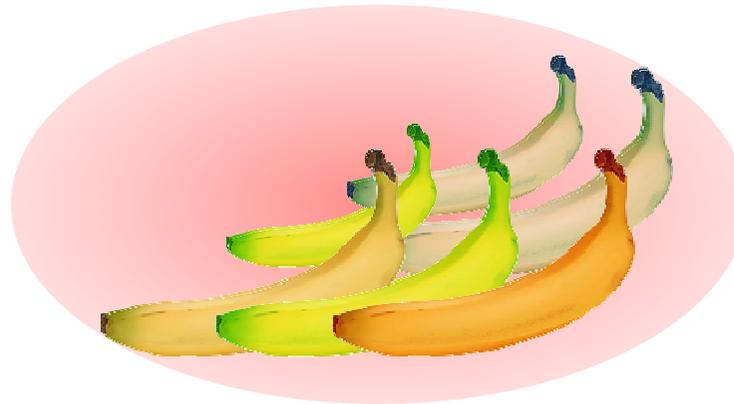
*dimensio* lernt die Inhalte...



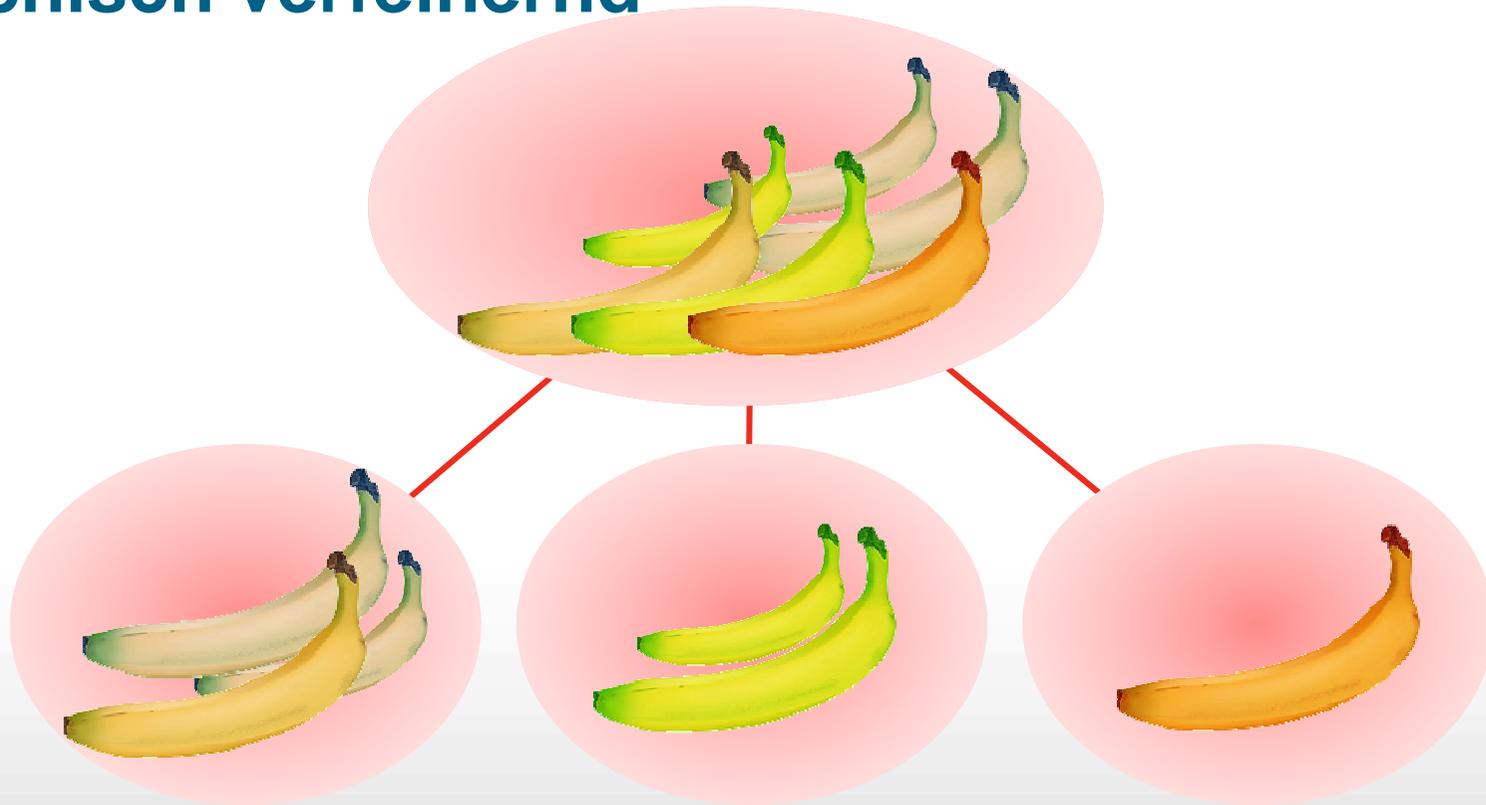
## ...und bildet Gruppen (Cluster)



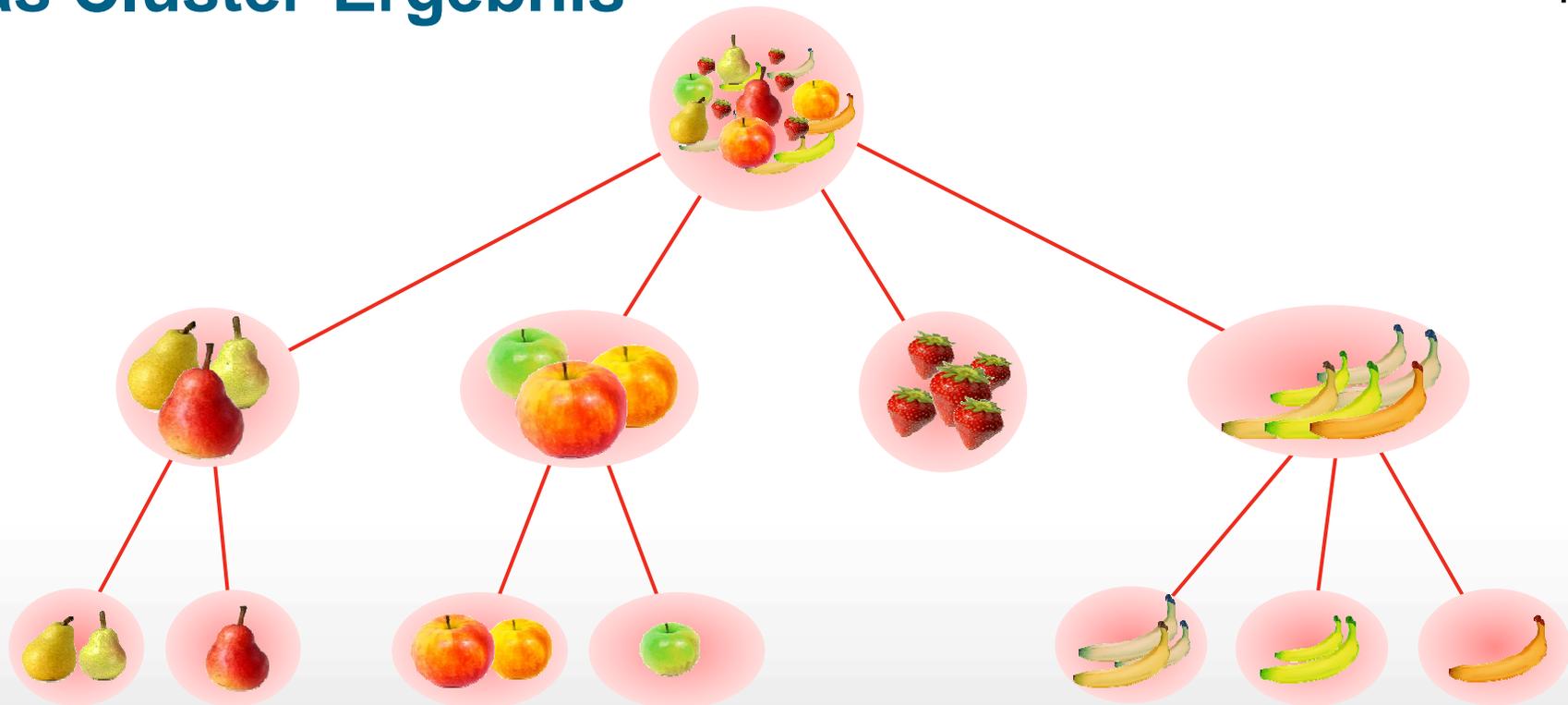
# Hierarchisch



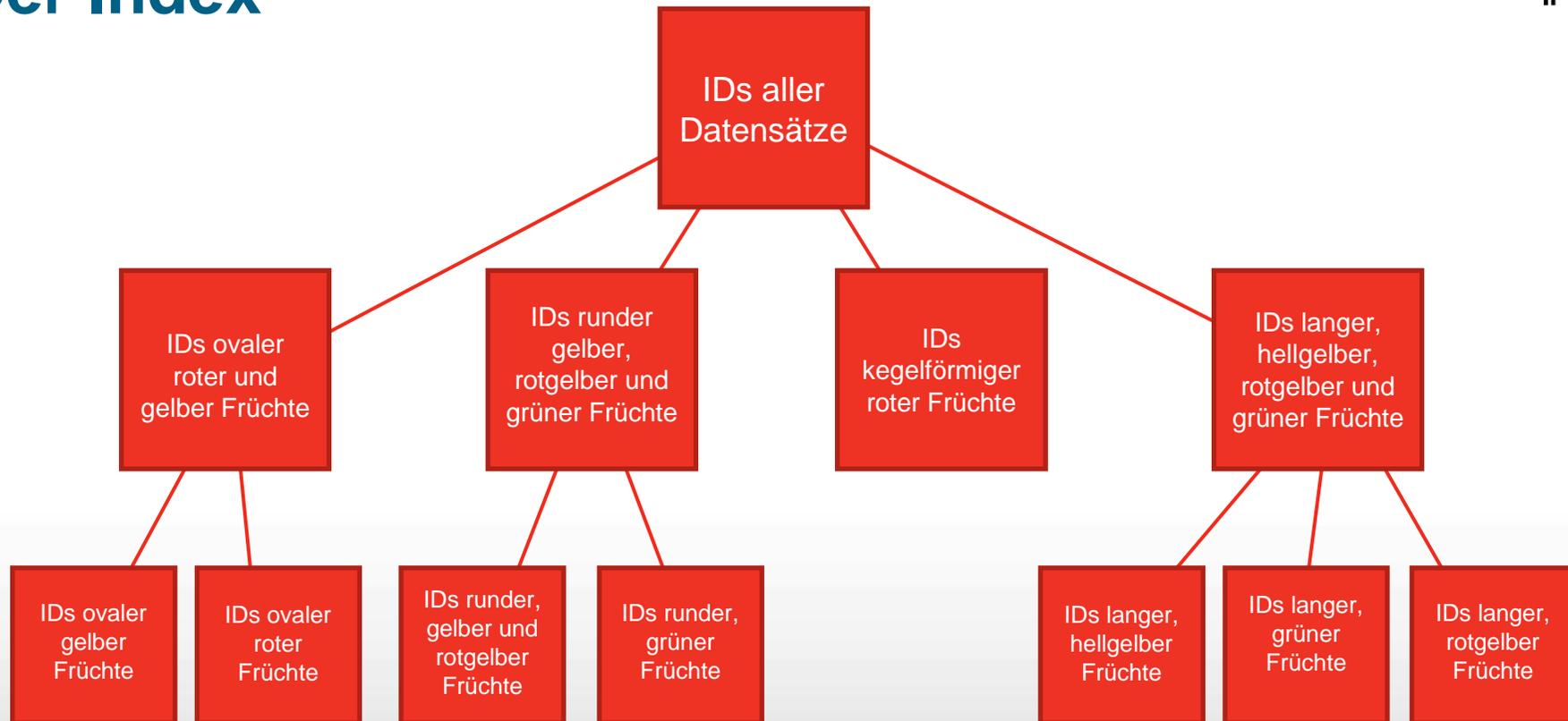
## Hierarchisch verfeinernd



# Das Cluster-Ergebnis

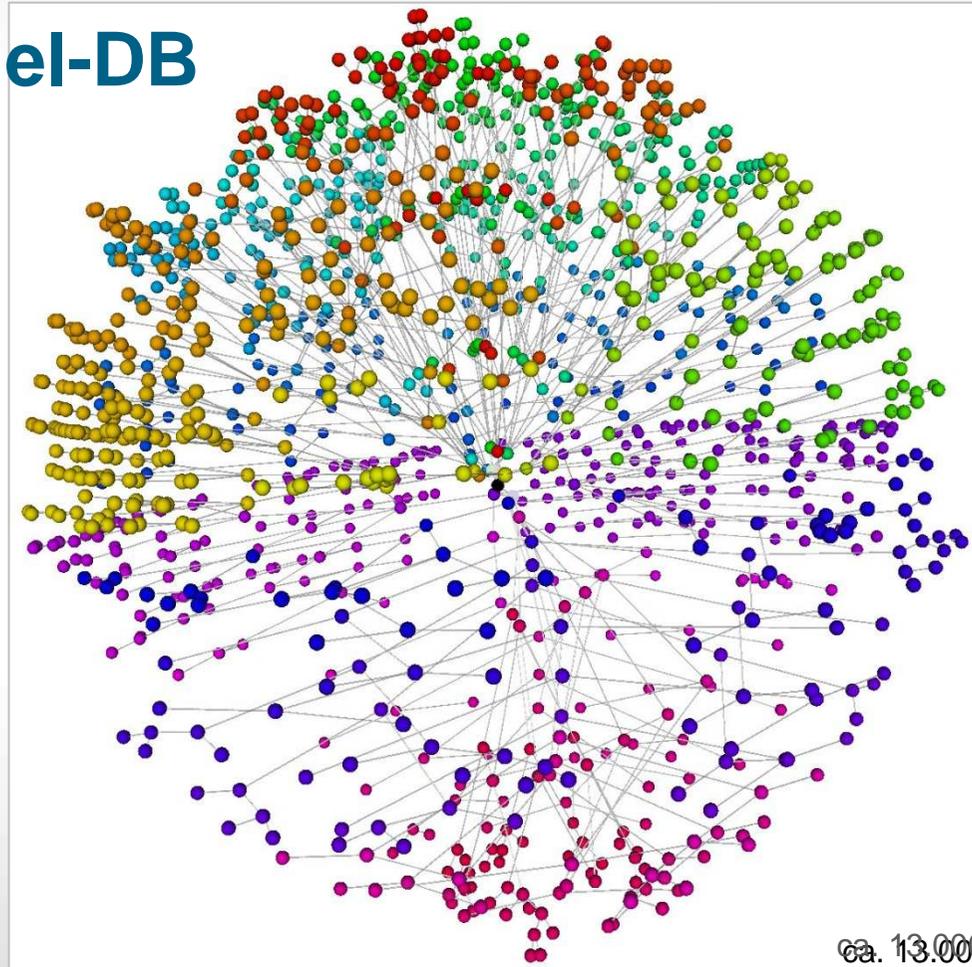


# Der Index





# Reale Beispiel-DB



ca. 13.000 Records mit 998 Dimensionen



# Performancegewinn



## Warum ist *dimensio* so schnell?

```
select *  
from Früchte  
where Form = rund  
or Form = lang  
and Farbe = grün
```

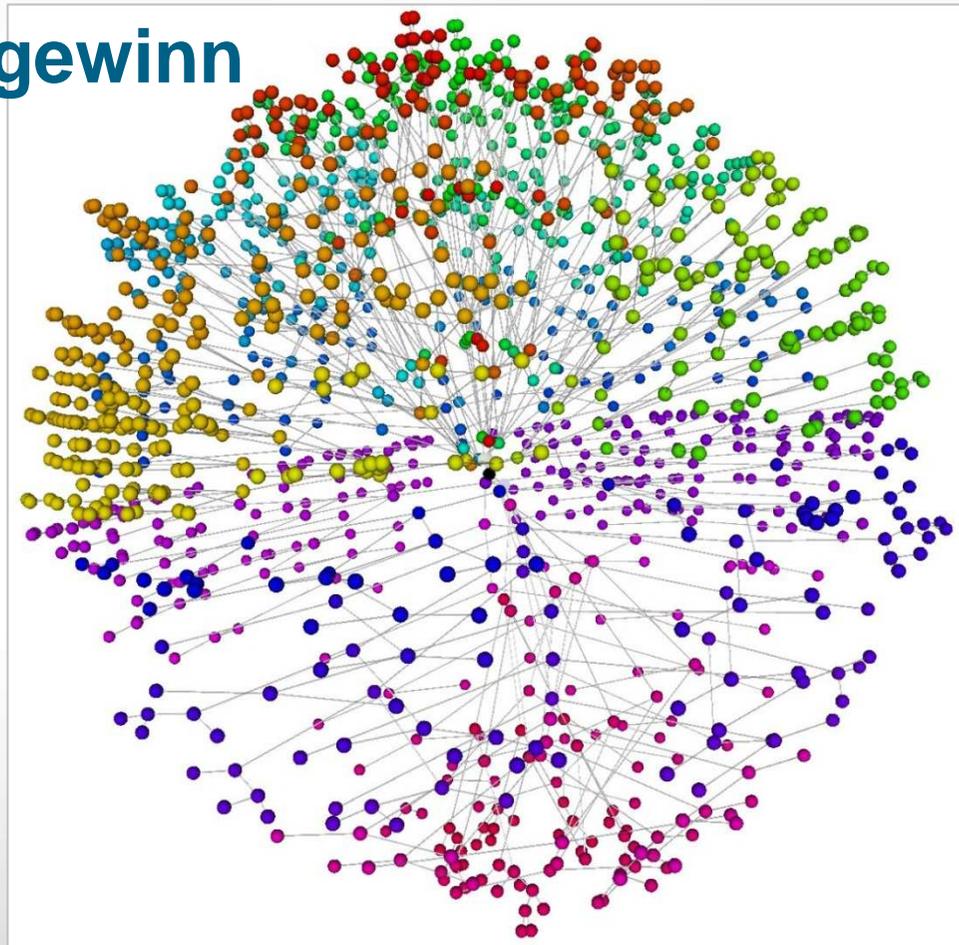
ohne *dimensio*:

Übergabe der Query an die DB,  
Prädikatauswertung,  
Selektion und Tupelrückgabe

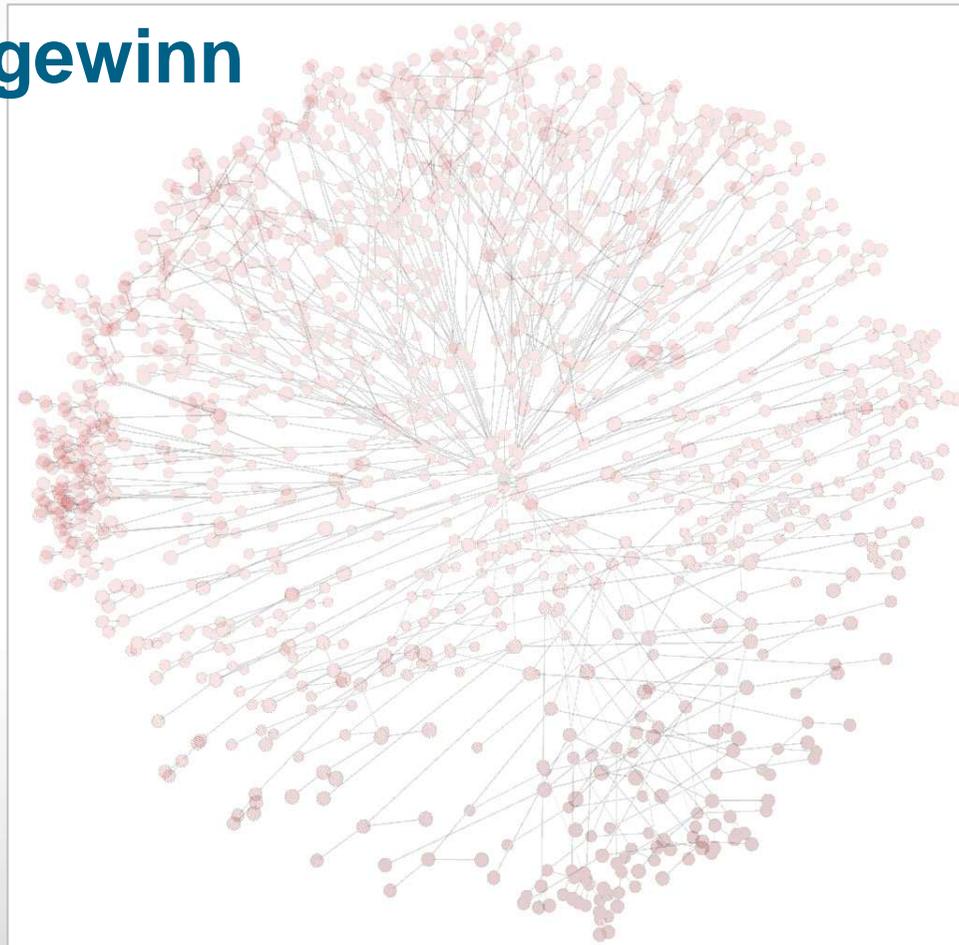
mit *dimensio*:

Übergabe der IDs an die DB,  
direkte Rückgabe der Tupel

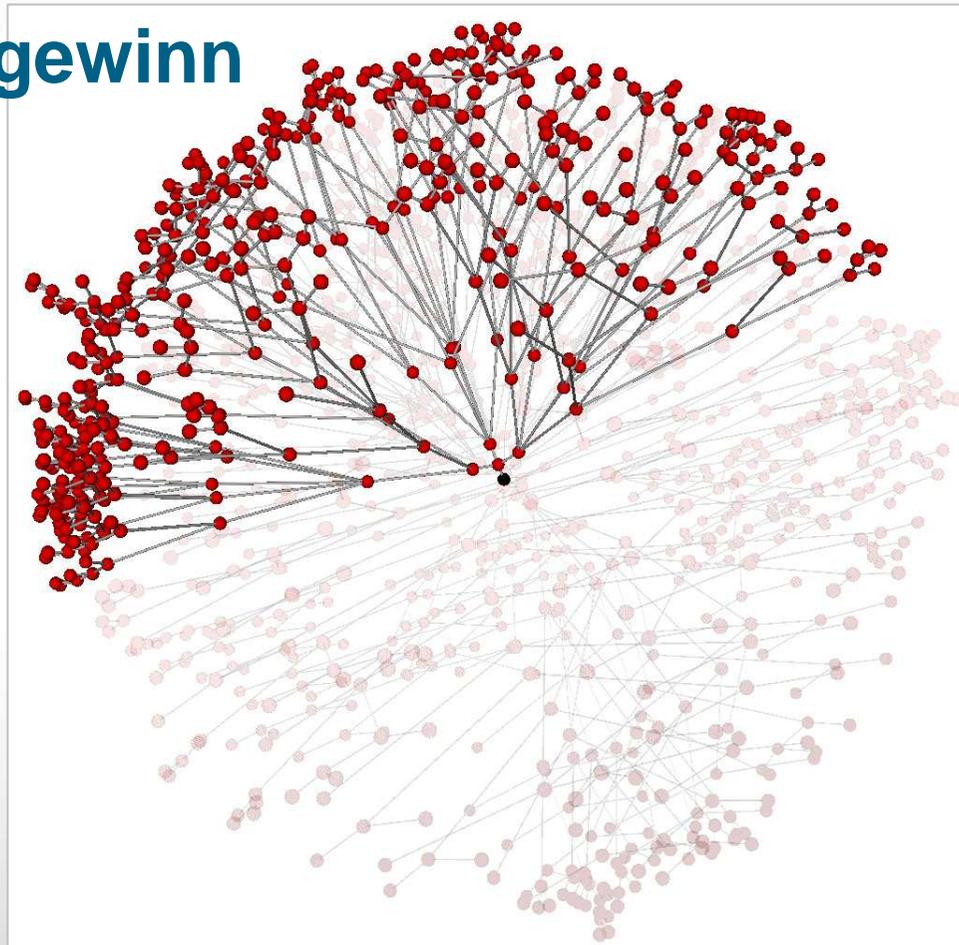
# Performancegewinn



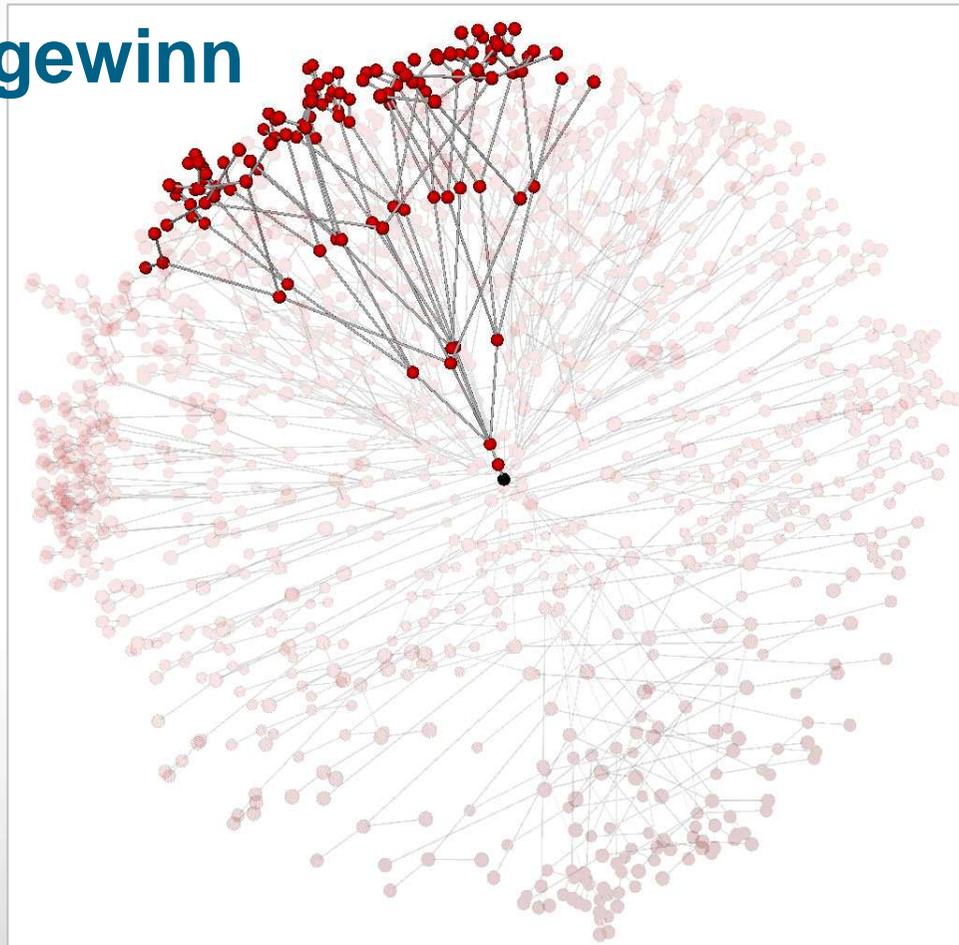
# Performancegewinn



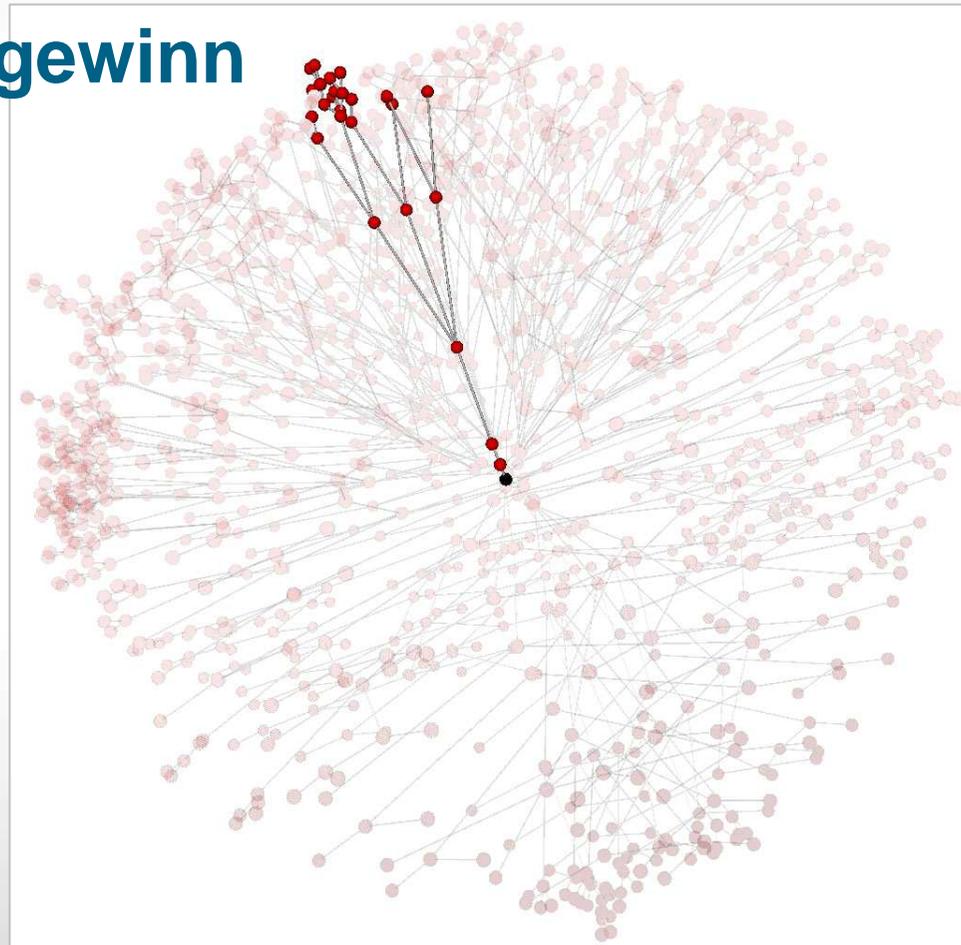
# Performancegewinn



# Performancegewinn

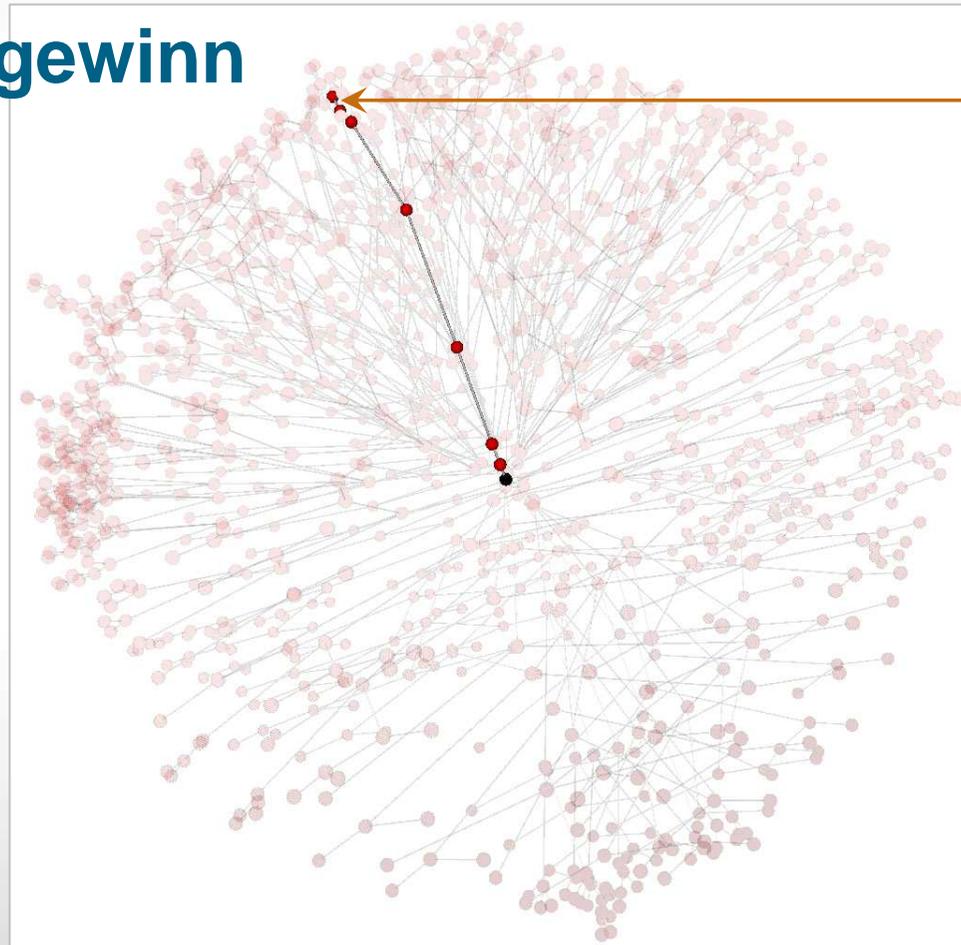


# Performancegewinn



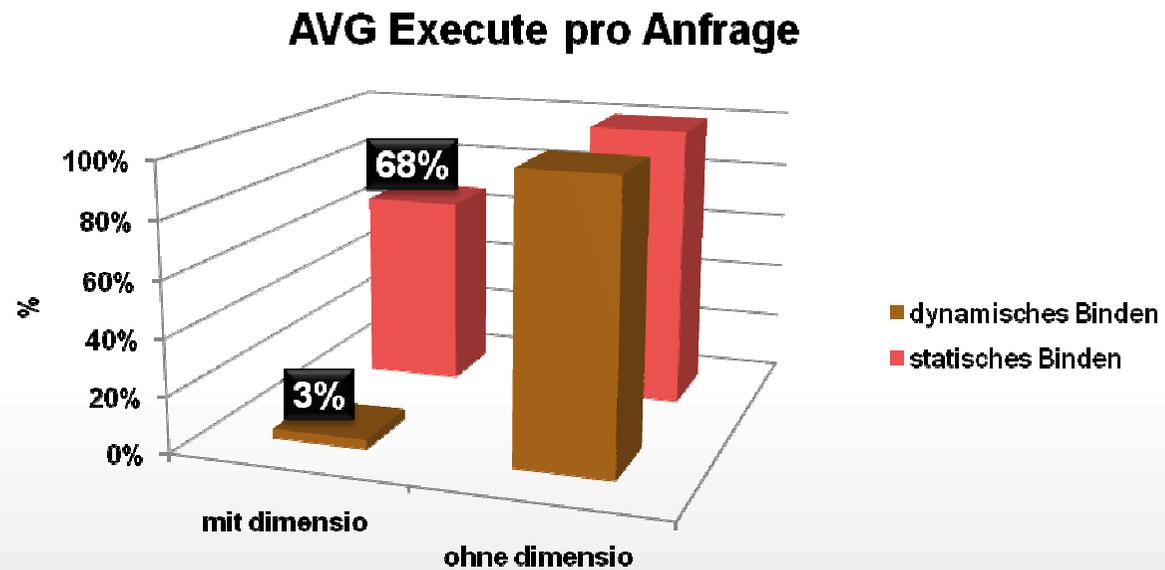


# Performancegewinn



ID
123
234
345
456
...
999

# Erster Test mit DB2





## Zweiter Test mit DB2

- Testumgebung:
  - Rechner: System x 3850 M2 (Klarabella) ; 4 x Quad-core Xeon X7350 2.93 GHz (8 MB L2 cache) ; 16 GB RAM ; 4 x 73 GB disks; Redhat 5.5 ; DB2 Enterprise 9.7 FP4
- Test-Datenbank:
  - bestehend aus 1 Tabelle mit 998 Spalten und ca. 8 Mio Datensätzen; Wertebereich: 0 oder 1
- Testscenario und verwendetes Tool:
  - Punktabfragen, Ausführung von insgesamt 10 – 20 Queries; DB2TOP
- DB2 Optimierung / DB2 Einstellungen:
  - Anzahl Bufferpool Pages wurden erhöht auf 400000 ; Datenbank wurde komprimiert, auf 1/10 der ursprünglichen Größe ; Index wurde komprimiert ; Setzen der kumulativen Registry-Variable DB2\_WORKLOAD= SAP ; Parallelität aktivieren durch Setzen des Parameters INTRA\_PARALLEL=YES
- Ergebnis:
  - **Die mit der *dimensio*-Software durchgeführten Abfragen auf die Datenbank waren 3x - 10x schneller als die Abfragen, die direkt an die DB2 geschickt wurden.**



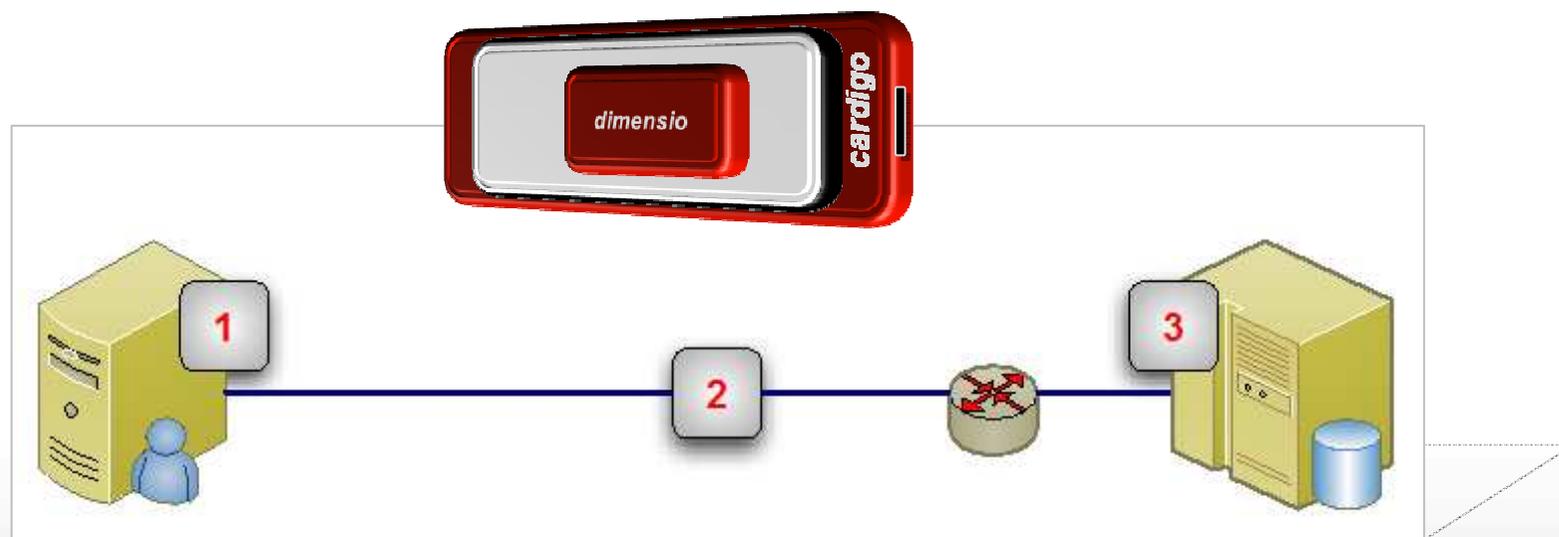
# Performancegewinn (PoC-Beispiele)

- Post-Processing
  - 4.000 Anfragen (Wochenleistung PKW-Produktion)
  - **6 Std.** ohne *dimensio* zu **3 Min. 20 Sec.** mit *dimensio*
  
- Bildvergleich
  - Ähnlichkeitssuche in 50.000 Bildern (Vergleich von Merkmalsvektoren)
  - **12 Sec.** ohne *dimensio* zu **0,001 Sec.** mit *dimensio*
  
- Business Intelligence
  - Anfragen mit ca. 2.000 Dimensionen (In-Memory + Column Store)
  - **20 Sec.** ohne *dimensio* zu **0,032 Sec.** mit *dimensio*



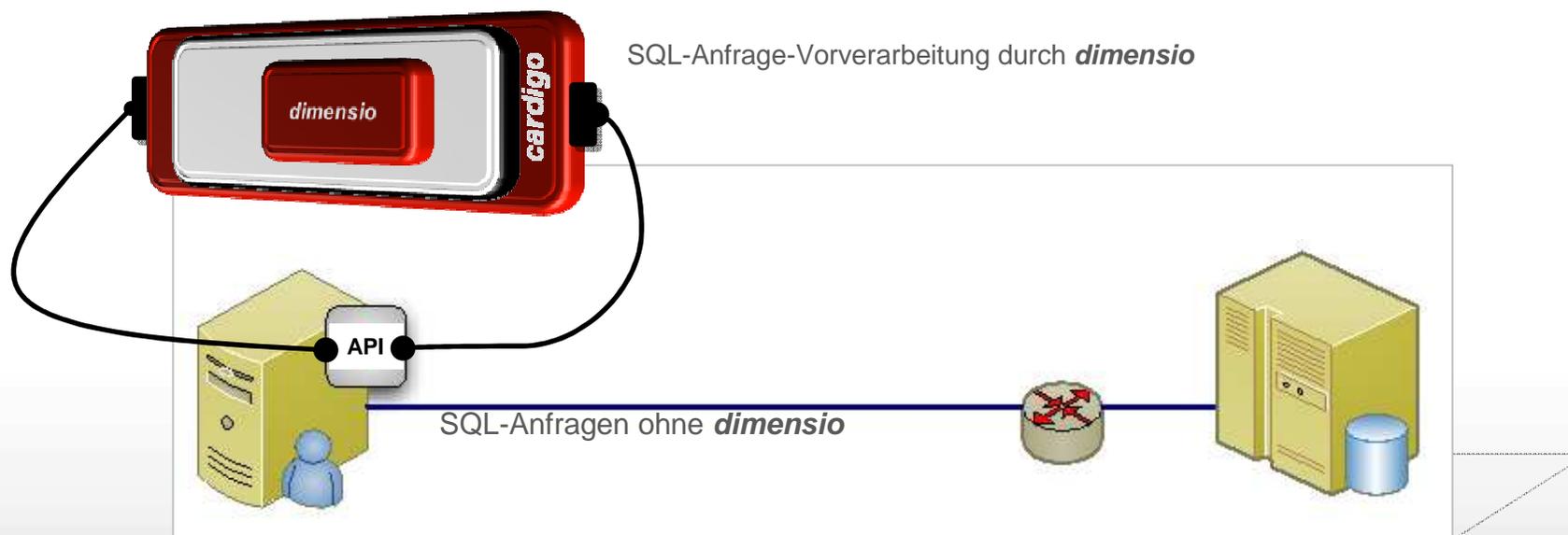
# Minimal-invasive System-Integration

# Minimal-Invasive Integration

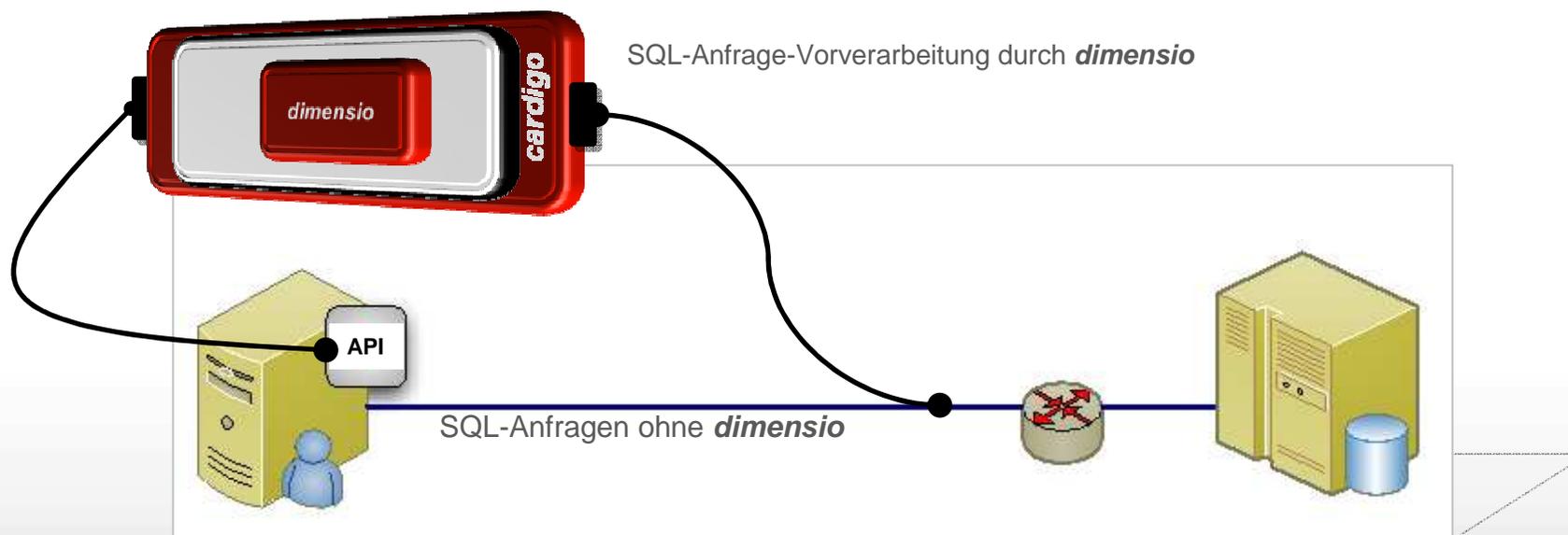


1. Integration in eine Applikation (*dimensio* Software Service)
2. Integration ins Netzwerk (*dimensio* Appliance / *dimensio* Virtual Appliance)
3. Integration in ein Datenbanksystem (Client als Plug-In, Server beliebig)

# Intransparente Integration

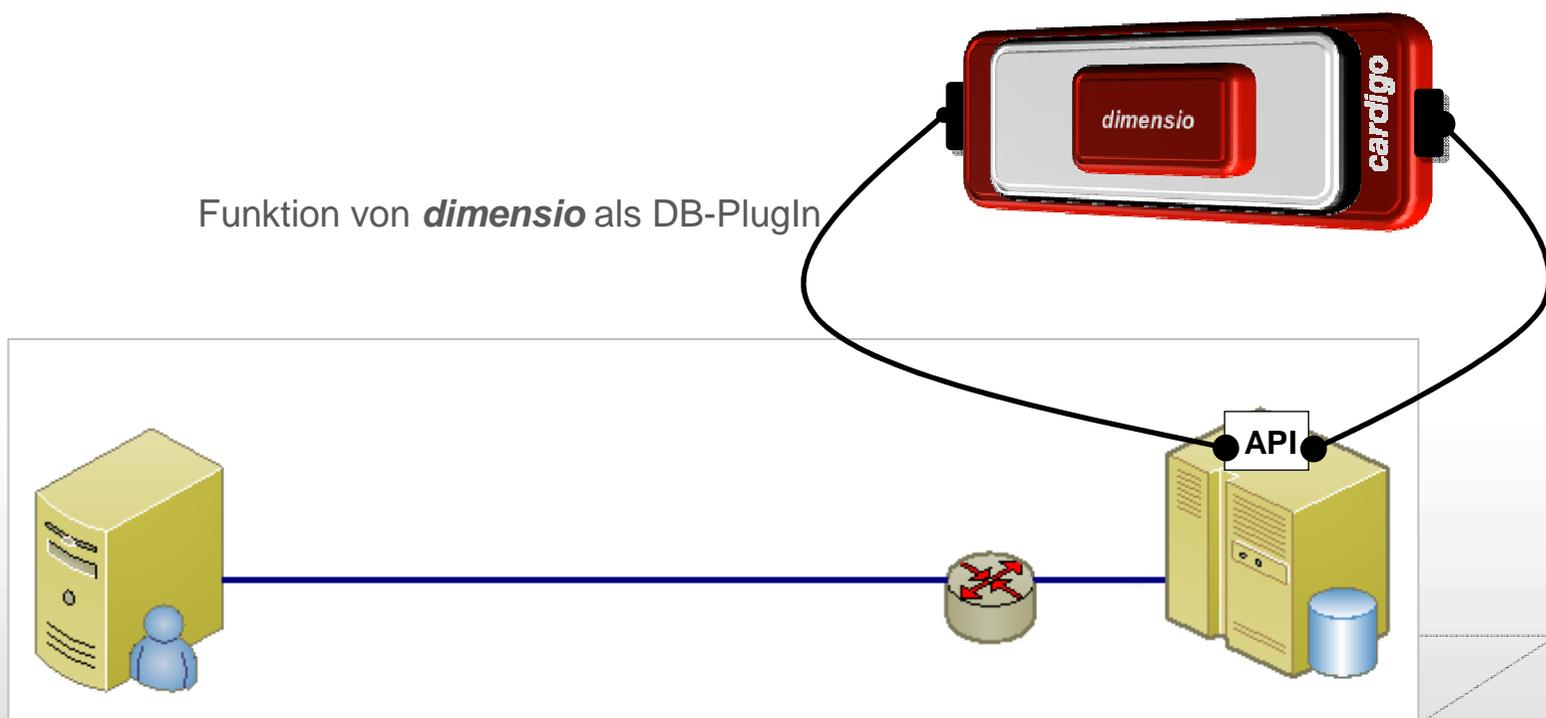


# Intransparente Integration



# Intransparente Integration

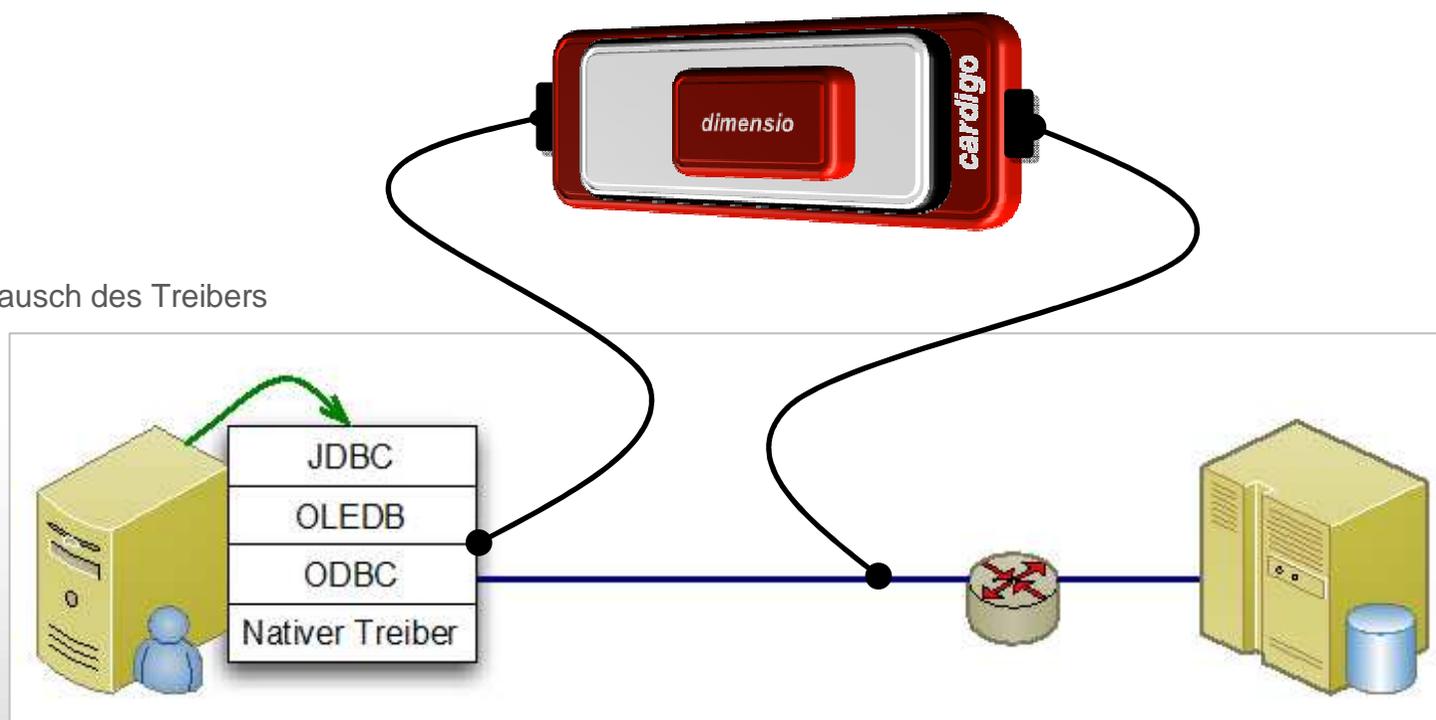
Funktion von *dimensio* als DB-PlugIn



# Transparente Integration

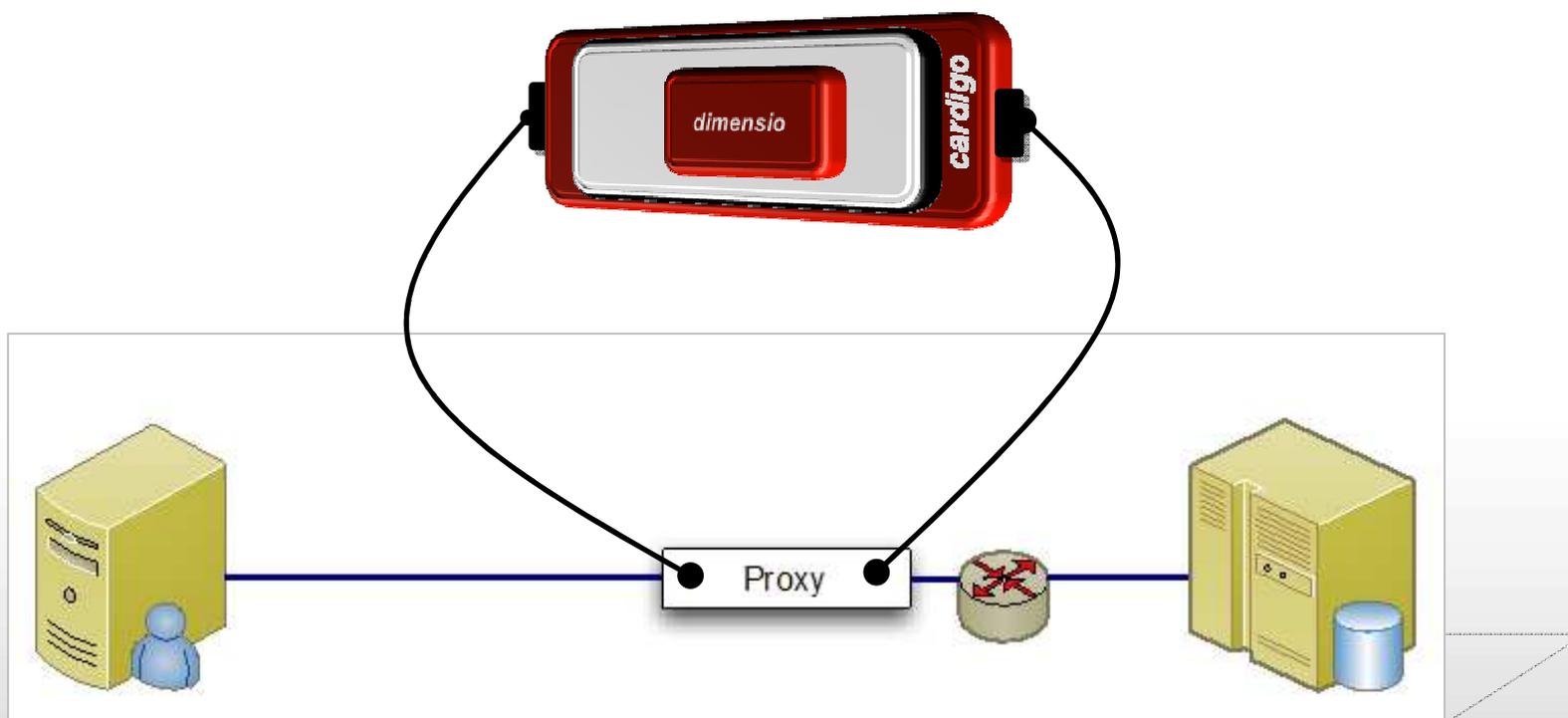
SQL-Anfrage-Vorverarbeitung durch *dimensio*;  
Anfragen ohne *dimensio* werden durchgeroutet

Austausch des Treibers



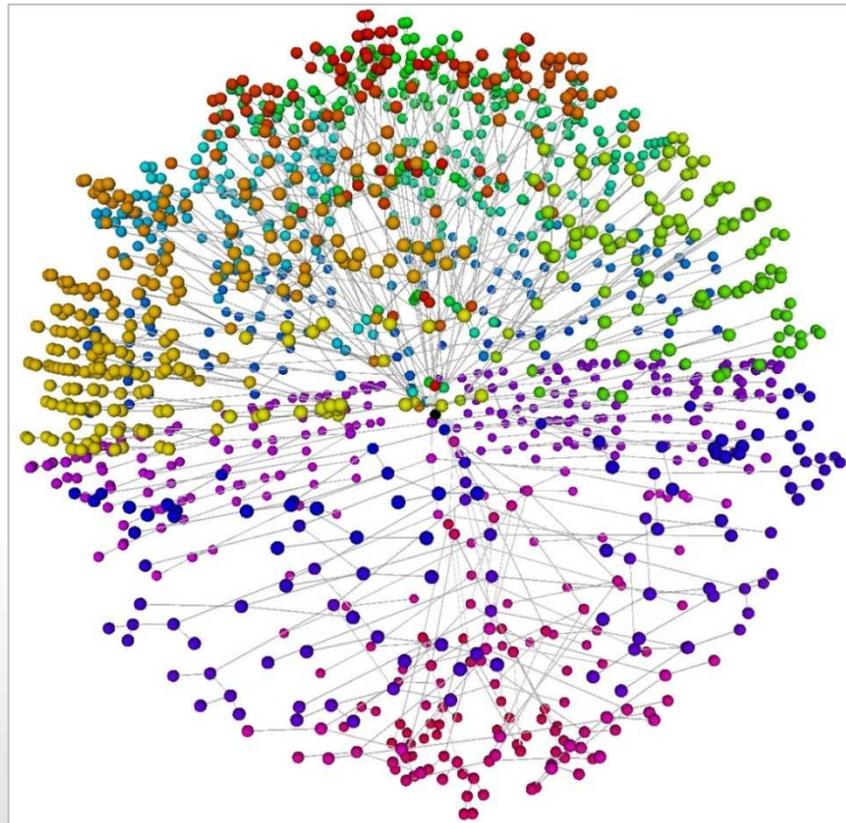
# Transparente Integration

SQL-Anfrage-Vorverarbeitung durch *dimensio*;  
Anfragen ohne *dimensio* werden durchgeroutet





## Danke für Ihre Aufmerksamkeit



Prof. Dr. Wolfgang Benn

Datenverwaltungssysteme  
Technische Universität Chemnitz  
D-09107 Chemnitz

dimensio informatics GmbH  
Brückenstr. 4  
D-09111 Chemnitz

Telefon: +49 371 26 20 19 0  
Telefax: +49 371 26 20 19 10

[info@dimensio-informatics.com](mailto:info@dimensio-informatics.com)  
[www.dimensio-informatics.com](http://www.dimensio-informatics.com)