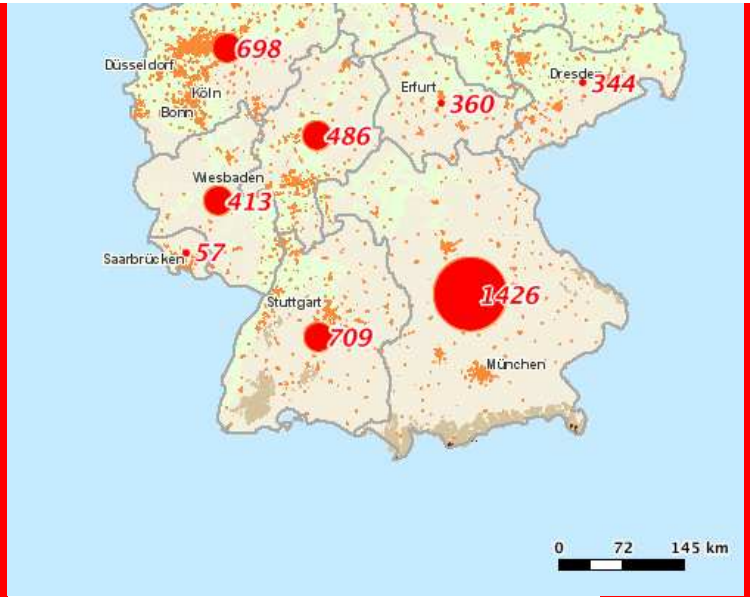
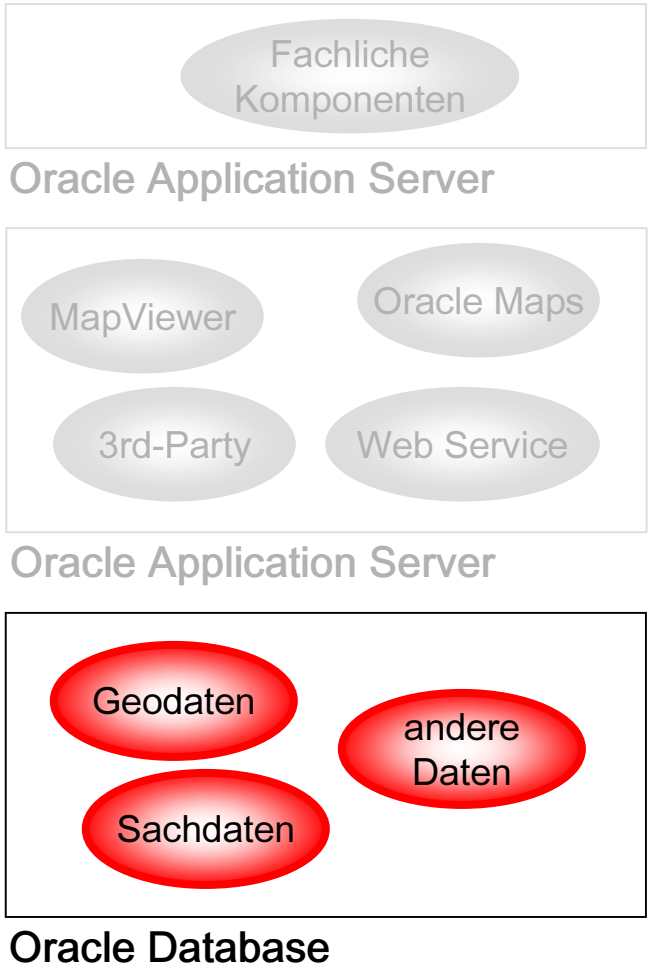


ORACLE®

Geodaten und die Oracle-Technologie
Oracle Locator, Spatial und Oracle MAPS

Das Fundament: Die Oracle-Datenbank

Offene Geodaten – offene Plattform!



Konkrete Lösungsapplikation
Fachliche Lösungen

Kartendarstellung
WMS Dienste
Routenberechnung

Geodaten / Räumliche Abfragen
Räumliche Indizes
Verschneidungen
Daten-Manipulationen

Oracle Spatial

Objektrelationale Implementierung

- Datentyp SDO_GEOMETRY

Tabelle Gemeinde

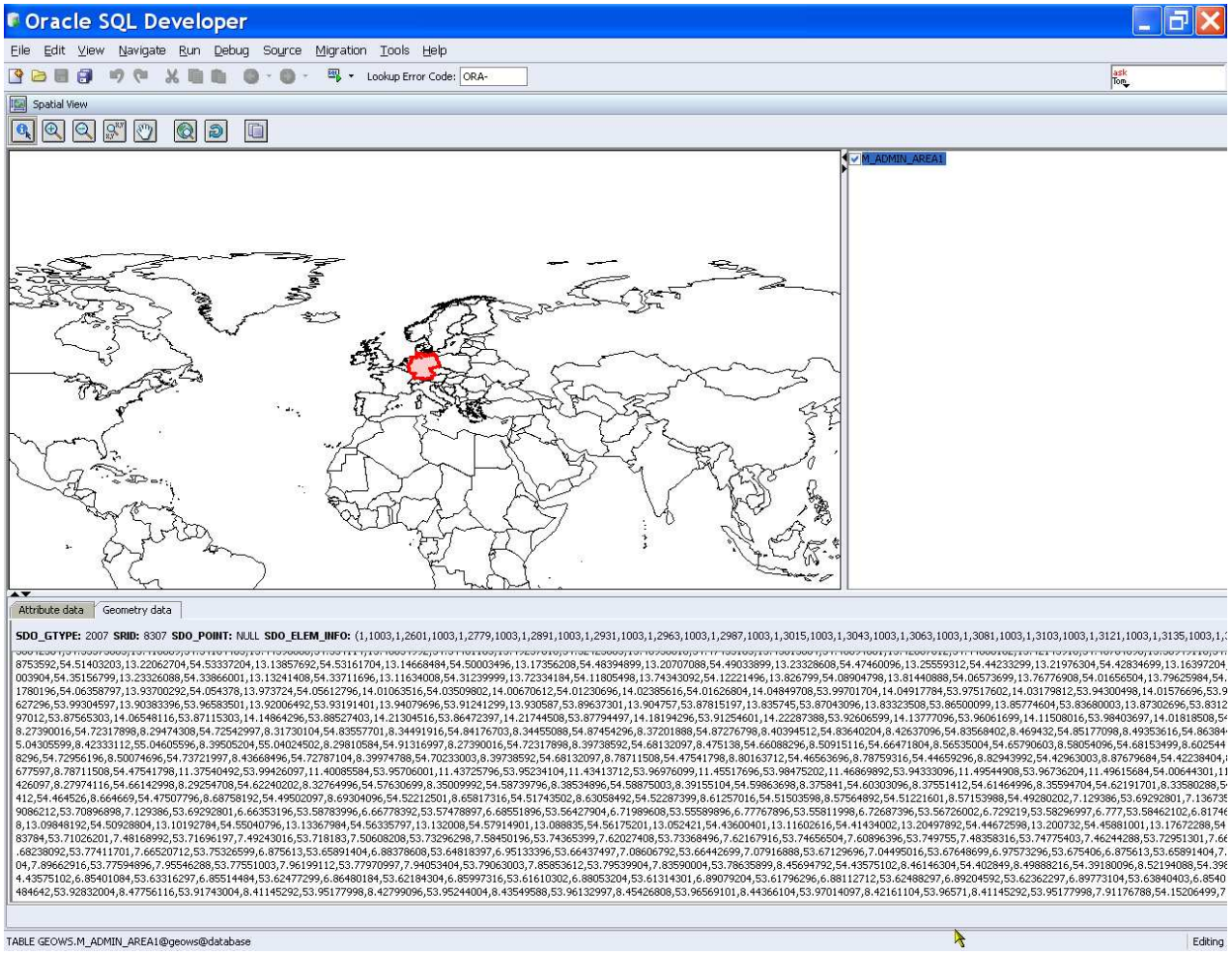
ID	NAME	GRENZE
NUMBER	VARCHAR2	SDO_GEOMETRY

SDO_GTYPE	NUMBER
SDO_SRID	NUMBER
SDO_POINT	SDO_POINT_TYPE
SDO_ELEM_INFO	SDO_ELEM_INFO_ARRAY
SDO_ORDINATES	SDO_ORDINATE_ARRAY

Datentyp SDO_GEOMETRY

- Nativer Datentyp für Geodaten
  - Vektordaten: SDO_GEOMETRY
  - Rasterdaten: SDO_GEORASTER
- Verwendbar wie jeder andere Datentyp
  - Tabellendefinitionen
  - PL/SQL-Logik
  - Constraints (eingeschränkt) und Trigger

Werkzeug:
GeoRaptor für SQL Developer



Datentyp SDO_GEOMETRY

Aufbau

```
SQL> describe sdo_geometry

 Name                                      Typ
 ----------------------------------------- ----------------------------
 SDO_GTYPE                                 NUMBER
 SDO_SRID                                  NUMBER
 SDO_POINT                                 MDSYS.SDO_POINT_TYPE
 SDO_ELEM_INFO                             MDSYS.SDO_ELEM_INFO_ARRAY
 SDO_ORDINATES                             MDSYS.SDO_ORDINATE_ARRAY

 MEMBER FUNCTION GET_GTYPE RETURNS NUMBER
 MEMBER FUNCTION GET_DIMS RETURNS NUMBER
 MEMBER FUNCTION GET_WKB RETURNS BLOB
 MEMBER FUNCTION GET_WKT RETURNS CLOB
 MEMBER FUNCTION ST_ISVALID RETURNS NUMBER
 :
```

Vektordaten "pur": OGC-Standards
Well Known Text und Well Known Binary

- GET()-Methoden ...

```
select e.geoloc.get_wkt() from de_hiway


E.GEOLOC.GET_WKT()
------------------------------------------
LINESTRING (9.18699984 47.77039998, 9.18969984
47.76949998, 9.19640016 47.763, 9.19919988
47.75840001, 9.20459988 47.75570001)

LINESTRING (9.24000012 47.73709998, 9.24130008
47.73349998, 9.23979996 47.72729997)
```

Geodaten-Tabellen erstellen

- SDO_GEOMETRY wie jeder andere Datentyp nutzbar
- Empfehlung: Nur eine SDO_GEOMETRY-Spalte

```
CREATE TABLE GEO_KONTINENTE(
       "AREA_ID"       NUMBER,
       "AREA_NAME"     VARCHAR2(20 BYTE),
       "FEATURE_TYPE" CHAR(9 BYTE),
       "GEOMETRY"      SDO_GEOMETRY
)
```

Spatial-Metadaten
SQL Developer

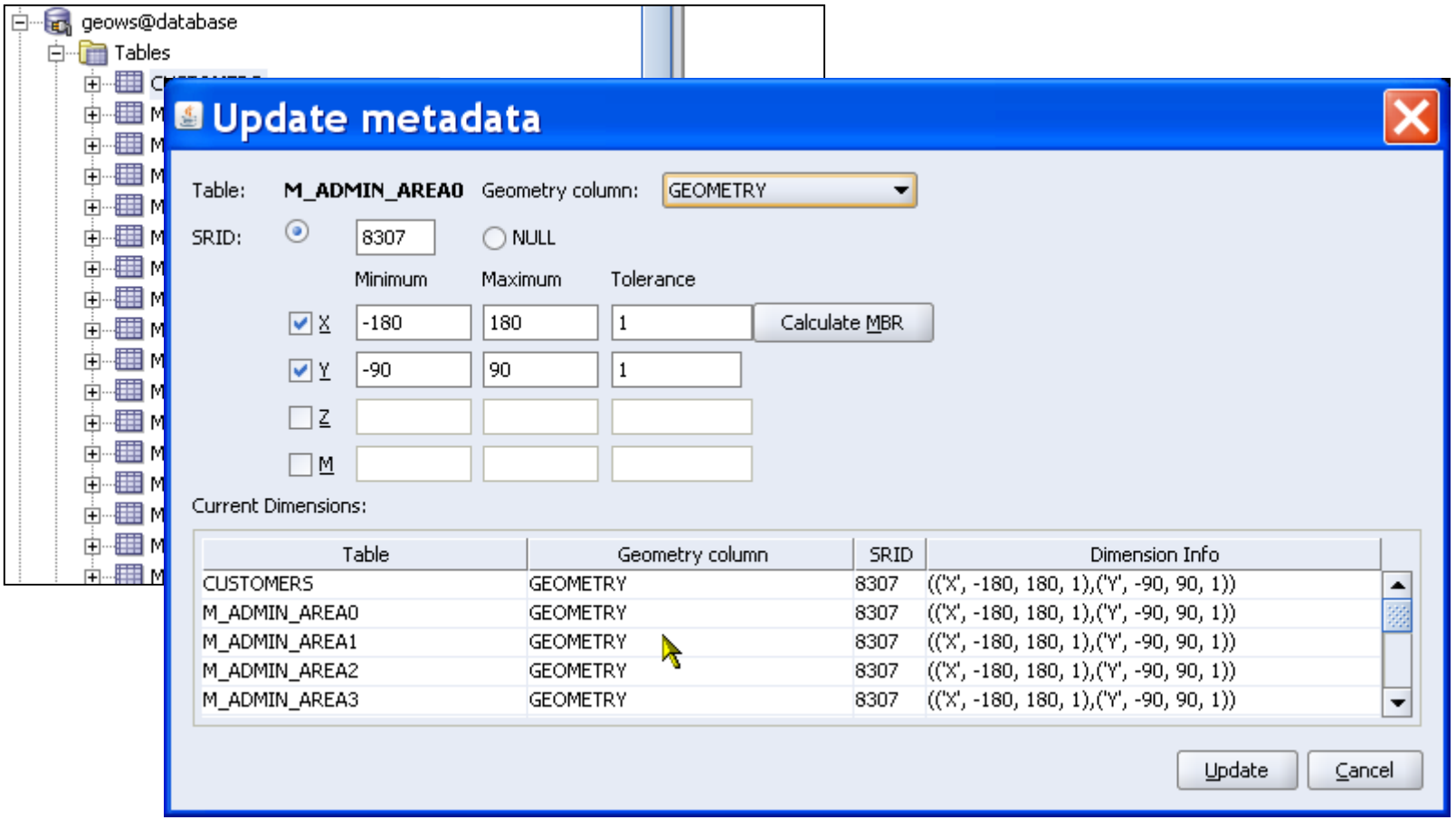


Tabelle am Beispiel: GEO_STAATEN

- Tabellenaufbau

```
SQL> desc geo_staaten

 Name                          Null?    Typ
 ----------------------------- -------- --------------------
 FEATURE_ID                             NUMBER
 FEATURE_NAME                           VARCHAR2(30)
 AREA_ID                                NUMBER
 CAPITAL                                VARCHAR2(30)
 FEATURE_TYPE                           CHAR(7)
 GEOMETRY                               MDSYS.SDO_GEOMETRY
```

Erste Schritte:
Der Aufbau von SDO_GEOMETRY

- Was sind die Koordinaten von Deutschland ...?

```
SQL> select id, x, y from
  2    geo_staaten,
  3    table(sdo_util.getvertices(geometry)) k
  4  where feature_name='GERMANY';

        ID           X           Y
---------- ---------- ----------
         1  14,720045  51,582406
         2 14,7554651  51,603937
         3 14,7570239  51,644129
         :          :          :
```

Erste Schritte:
Der Aufbau von SDO_GEOMETRY

- Welches Koordinatensystem wird verwendet ...?

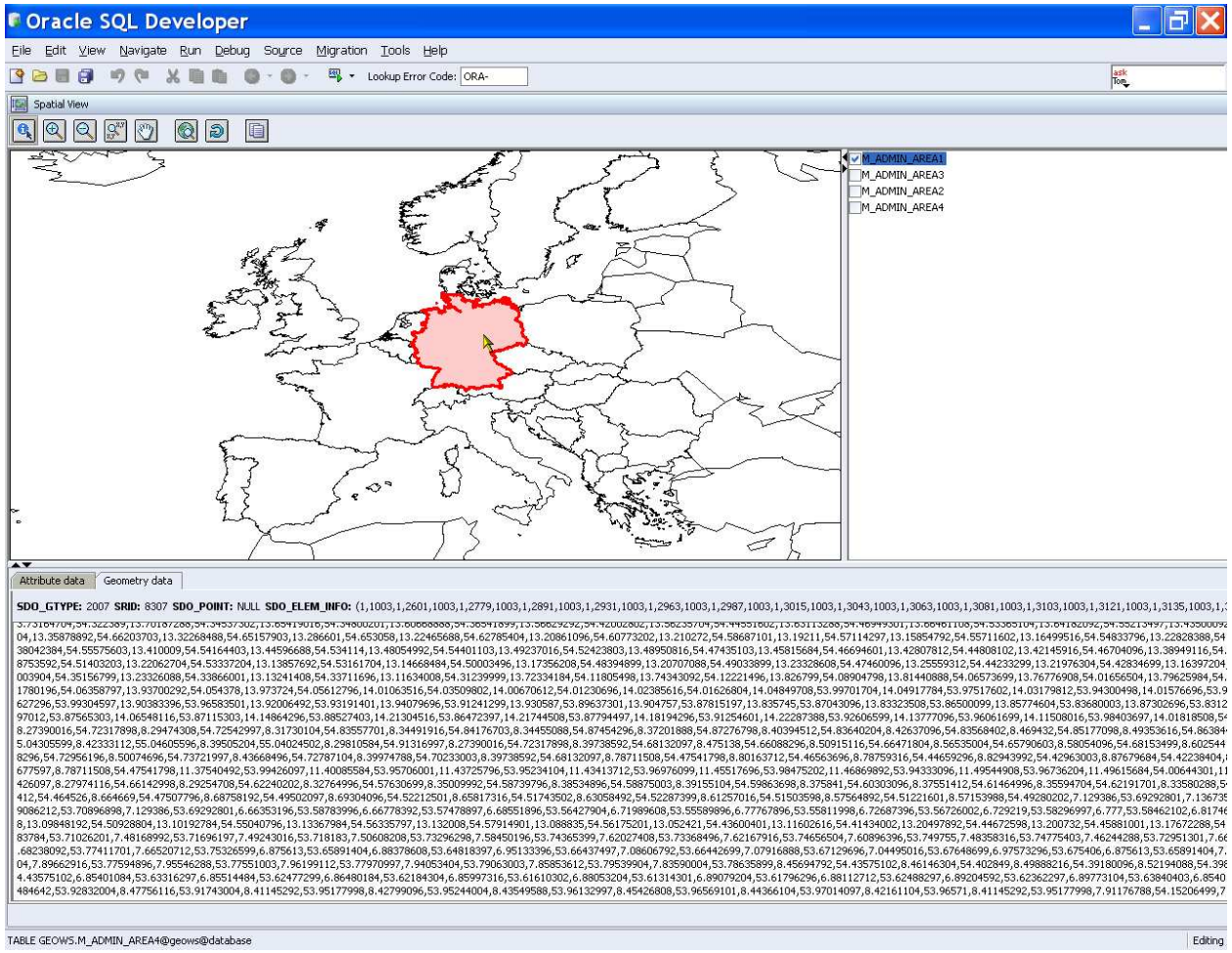
```
SQL> select e.geometry.sdo_srid
  2  from geo_staaten e
  3  where feature_name='GERMANY';

GEOMETRY.SDO_SRID
-----------------
             8307

SQL> select cs_name from cs_srs where srid=8307;

CS_NAME
-------------------------------------------------
Longitude / Latitude (WGS 84)
```

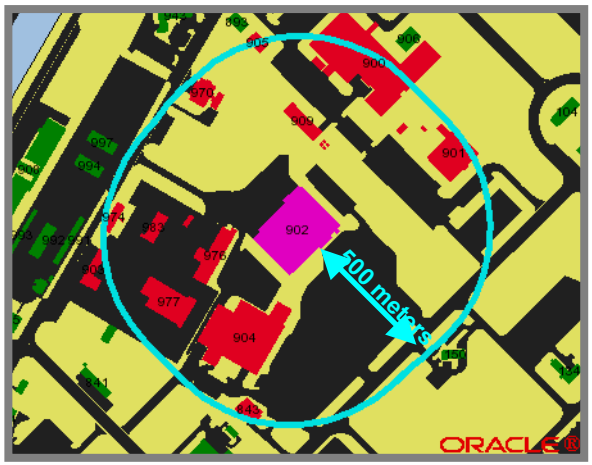
Und im SQL Developer



Oracle Spatial
SQL Operatoren

- Enge SQL-Integration
  - "Topologische Operatoren"
    - Inside Contains
    - Touch Disjoin
    - Covers Covered By
    - Equal Overlap Boundary
  - "Dinstanz-Operatoren"
    - Within Distance
    - Nearest Neighbor

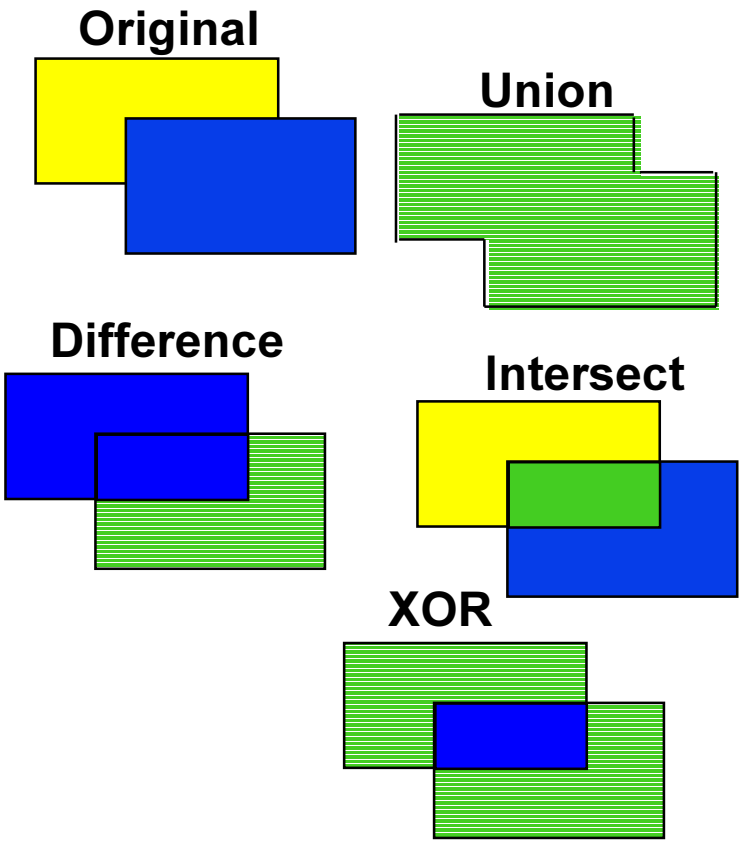
Inside



Within Distance

Oracle Spatial
SQL Funktionen

- Rückgabe: SDO_GEOMETRY
  - Union
  - Difference
  - Intersect
  - XOR
  - Buffer
  - Centroid
  - ConvexHull
- Rückgabe: NUMBER
  - Length
  - Area
  - Distance
  - Volume



Geometrische Operationen

- Wie groß ist Deutschland ...?

```
SQL> select sdo_geom.sdo_area(
  2     geometry, 1, 'unit=sq_km'
  3  ) "Fläche km²"
  4  from geo_staaten e
  5  where feature_name='GERMANY';

 Fläche km²
-----------
 356644,105

1 Zeile wurde ausgewählt.
```

Einheit (km)

Kombinierte Abfragen

- Welche Kunden mit Umsatz sind in Brandenburg?

```
SQL> select name, umsatz
  2  from customers c, geo_laender gl
  3  where sdo_relate(
  4    c.geometry, gl.geometry, 'mask=anyinteract'
  5  ) = 'TRUE'
  6  and gl.feature_name='Brandenburg'
  7  and umsatz is not null and umsatz > 0

NAME                                 UMSATZ
------------------------------ ----------
Freie Tanke                            4876
Marhoul                                4798
Tankstelle Schönefeld                  4598
Tankstelle Dreieck Havelland           7896
:                                         :
```

Auswertungen

- Bundesländer nach Umsatz ...

```
SQL> select
  2    gl.feature_name, sum(c.umsatz) umsatz_gesamt
  3  from customers c, geo_laender gl
  4  where sdo_relate(
  5    c.geometry, gl.geometry, 'mask=anyinteract'
  6  ) = 'TRUE'
  7  and umsatz is not null and umsatz > 0
  8  group by gl.feature_name

FEATURE_NAME                       UMSATZ_GESAMT
---------------------------------- -------------
Bayern                                      5000
Berlin                                     77116
Brandenburg                                62447
Niedersachsen                                500
:                                              :
```

Sonderfall "Nächster Nachbar"

- Die nächsten Nachbarn von "Carsten Czarski"

```
SQL> select
  2    c1.name, c1.umsatz, sdo_nn_distance(1) dist_km
  3  from customers c1, customers c2
  4  where c2.name = 'Carsten Czarski'
  5  and sdo_nn(
  6    c1.geometry,
  7    c2.geometry,
  8    'sdo_batch_size=0 unit=km',
  9    1
 10  ) = 'TRUE
 11  and c1.umsatz is not null
 12  and c1.name !='Carsten Czarski'
 13  and rownum<=5
```

Sonderfall "Nächster Nachbar"

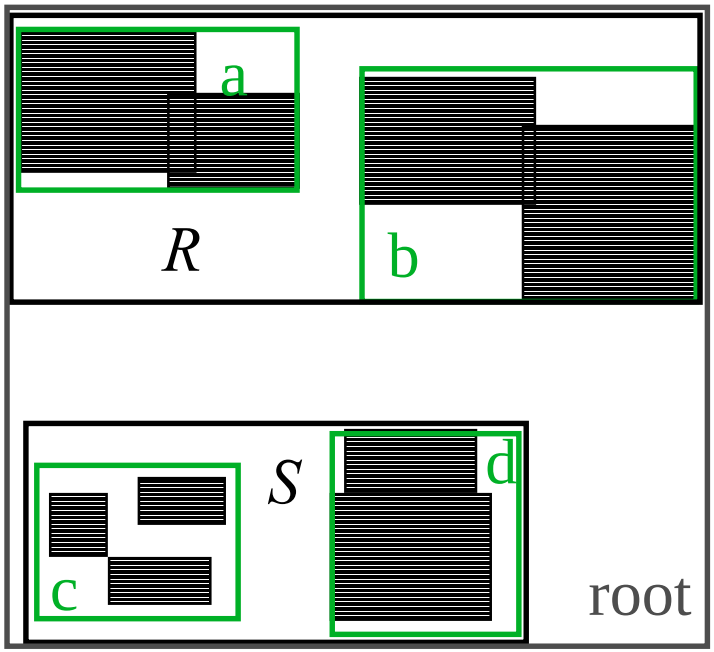
- Die nächsten Nachbarn von "Carsten Czarski"

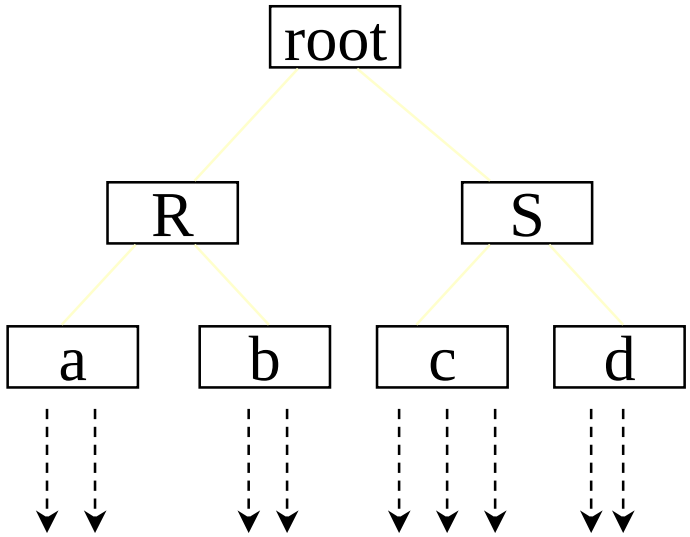
NAME	UMSATZ	DIST_KM
Der Säubrenner	100	2,62556936
Michael Künzner	2300	8,45149714
Eifel Power II	2000	8,61981335
Jörn Huber	4000	9,41099027
Mike Müller	0	15,8498991

5 Zeilen ausgewählt.

Spatial Index

Aufbau des R-Tree-Index

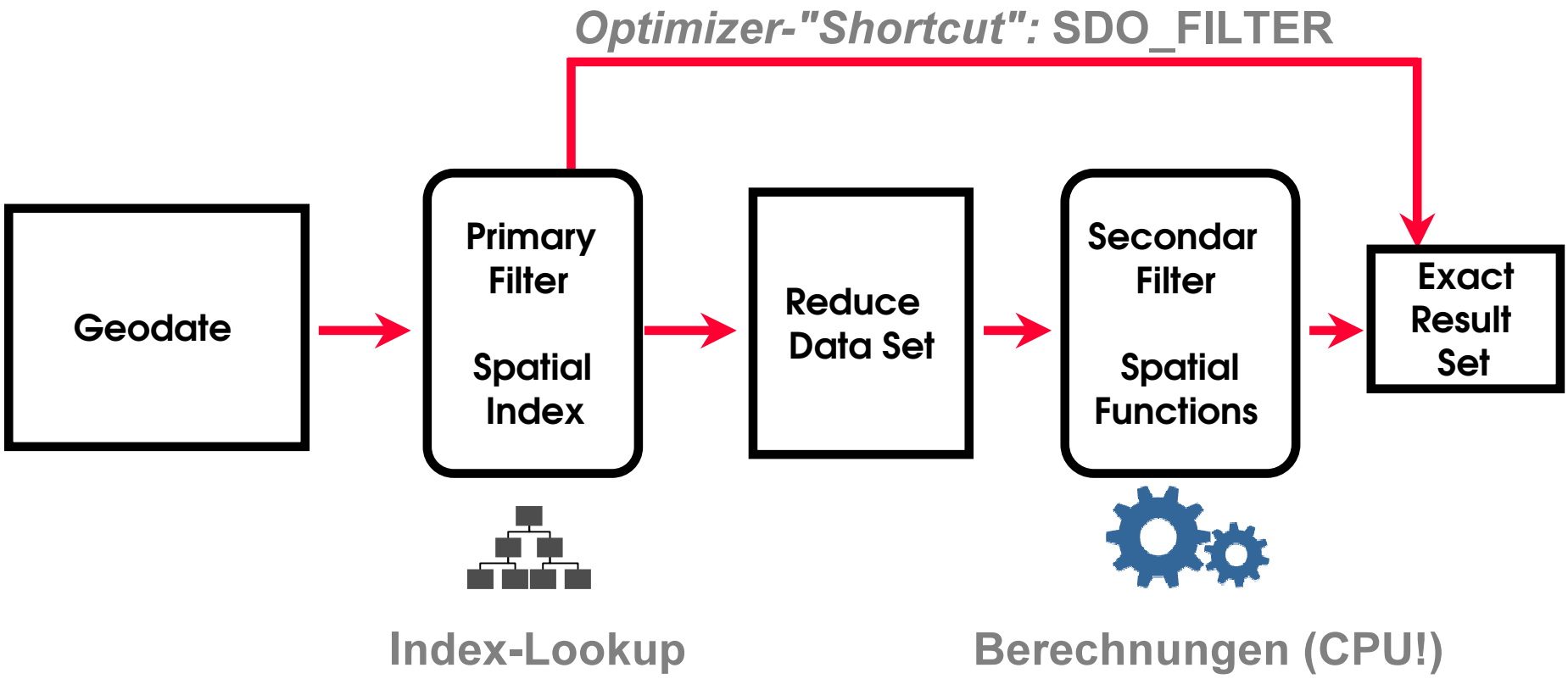




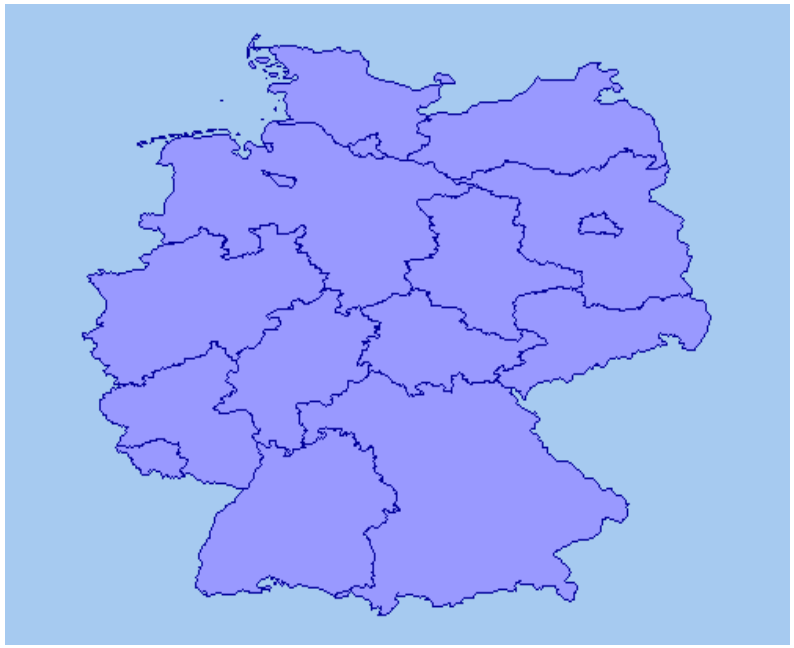
Zeiger auf Geometrien

Nutzung des Spatial-Index

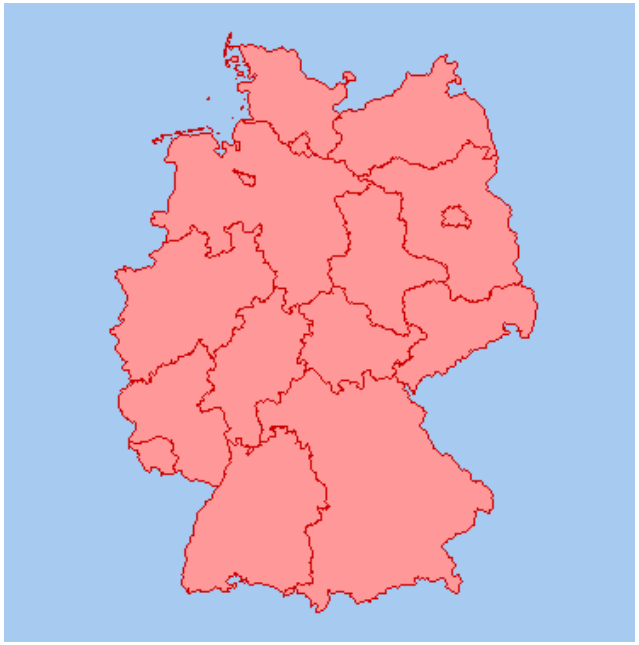
Primär- und Sekundärfilter



Koordinatensysteme



(Längen/Breitengrade)



Projektion

Koordinatensysteme: EPSG

- Datenmodell für CS-Parameter
- SRID ist standardisiert

CS	EPSG_ID	SDO_ID
WGS84	4326	8307
ETRS89	25832	8220
GK 2 (Bessel)	31466	82015
GK 3 (Bessel)	31467	82027
GK 4 (Bessel)	31468	82032
GK 5 (Bessel)	31469	82034
UTM Zone 32N (WGS84)	32632	82344
:	:	:

EPSG-Koordinatensysteme
Anwendungsfall WMS

- REQUEST=getCapabilities

```
- <Layer queryable="0" opaque="0" noSubsets="0">
    <Name>Germany</Name>
    <Title>Germany</Title>
    <SRS>epsg:4326</SRS>
    <SRS>epsg:31467</SRS>
    <SRS>epsg:31466</SRS>
    <SRS>epsg:31468</SRS>
    <SRS>epsg:31469</SRS>
    <SRS>epsg:31492</SRS>
    <SRS>epsg:31493</SRS>
    <SRS>epsg:31494</SRS>
    <SRS>epsg:31495</SRS>
    <SRS>epsg:31462</SRS>
    <SRS>epsg:31463</SRS>
    <SRS>epsg:31464</SRS>
    <SRS>epsg:31465</SRS>
    <SRS>epsg:25832</SRS>
    <SRS>epsg:25833</SRS>
    <SRS>epsg:31287</SRS>
    <SRS>epsg:32633</SRS>
    <SRS>epsg:2398</SRS>
    <LatLonBoundingBox minx="5.60075" miny="47.2441" maxx="15.425"
```

EPSG-Koordinatensysteme

- Nicht nur die SRID ist standardisiert ...

8307 Longitude / Latitude (WGS 84)

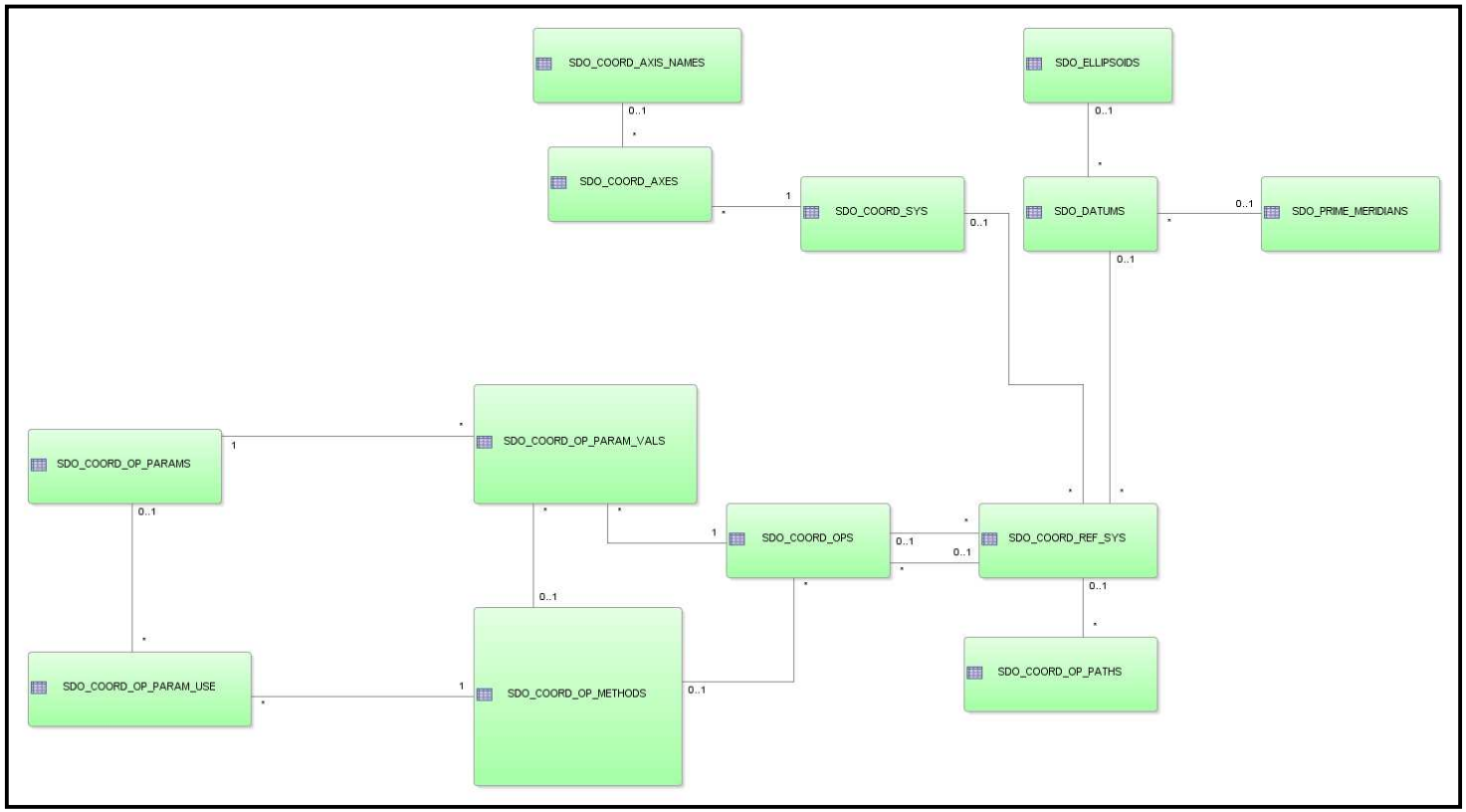
```
GEOGCS [ "Longitude / Latitude (WGS 84)", DATU
M ["WGS 84", SPHEROID ["WGS 84", 6378137, 298.
257223563]], PRIMEM [ "Greenwich", 0.000000 ],
UNIT ["Decimal Degree", 0.0174532925199433O]]
```

4326 WGS 84

```
GEOGCS [ "WGS 84", DATUM ["World Geodetic Syst
em 1984 (EPSG ID 6326)", SPHEROID ["WGS 84 (EP
SG ID 7030)", 6378137, 298.257223563]], PRIMEM
[ "Greenwich", 0.000000 ], UNIT ["Decimal Degr
ee", 0.0174532925199433O]]
```

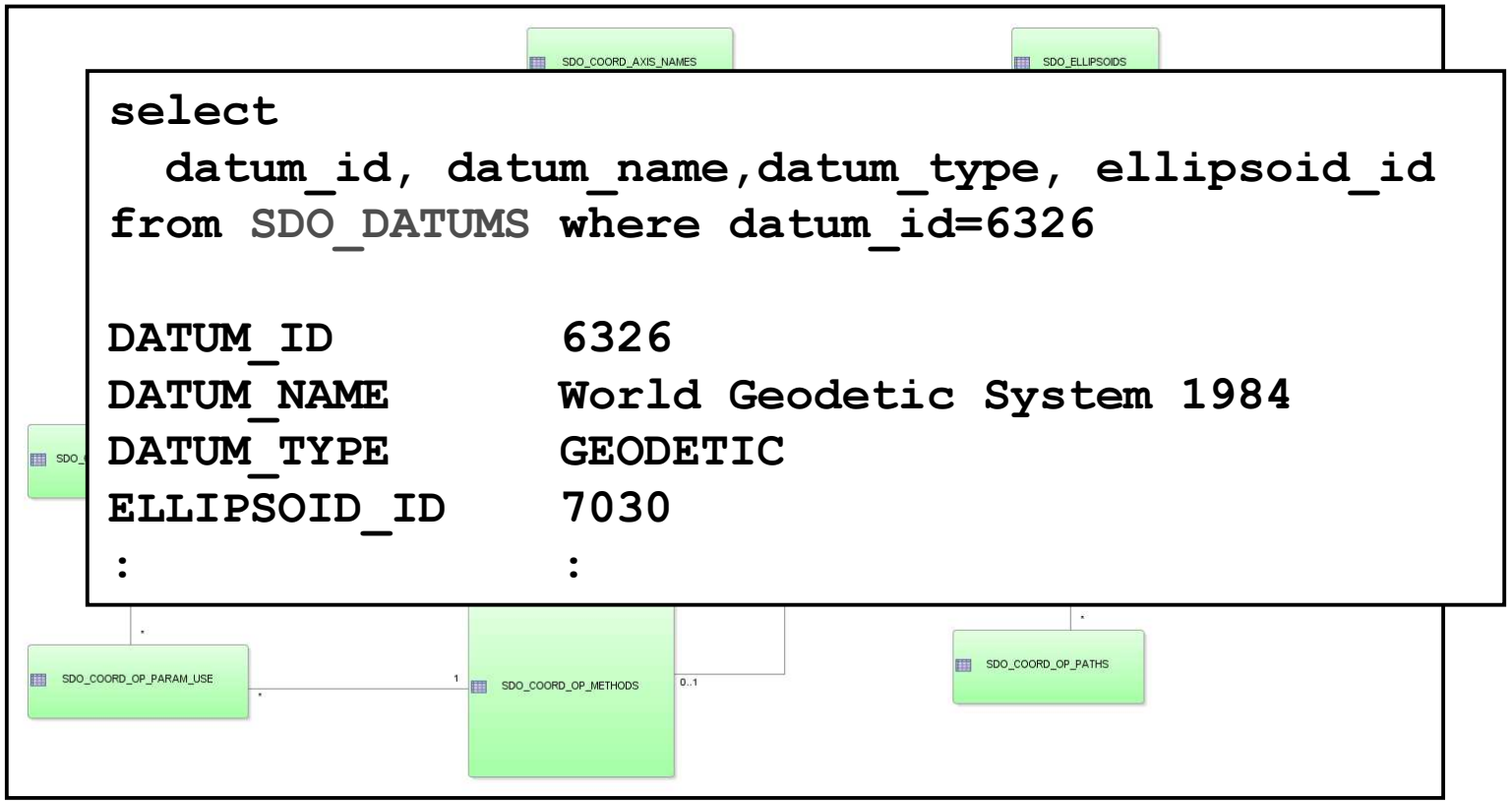
EPSG-Komponenten

- EPSG-Datenmodell (Auszug)



EPSG-Komponenten

- EPSG-Datenmodell (Auszug)

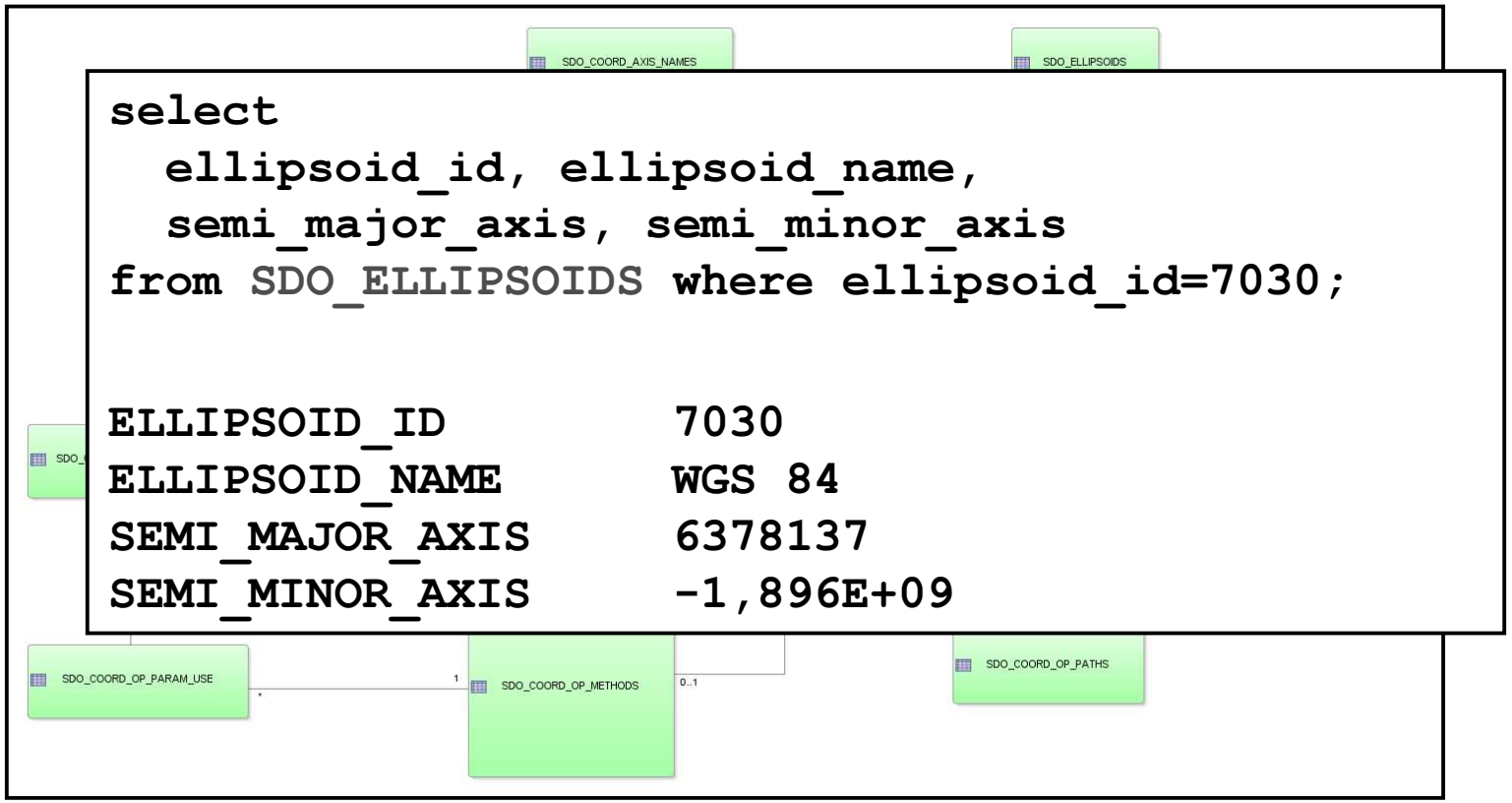



```
select
  datum_id, datum_name,datum_type, ellipsoid_id
from SDO_DATUMS where datum_id=6326

DATUM_ID          6326
DATUM_NAME        World Geodetic System 1984
DATUM_TYPE        GEODETIC
ELLIPSOID_ID      7030
:                 :
```

EPSG-Komponenten

- EPSG-Datenmodell (Auszug)

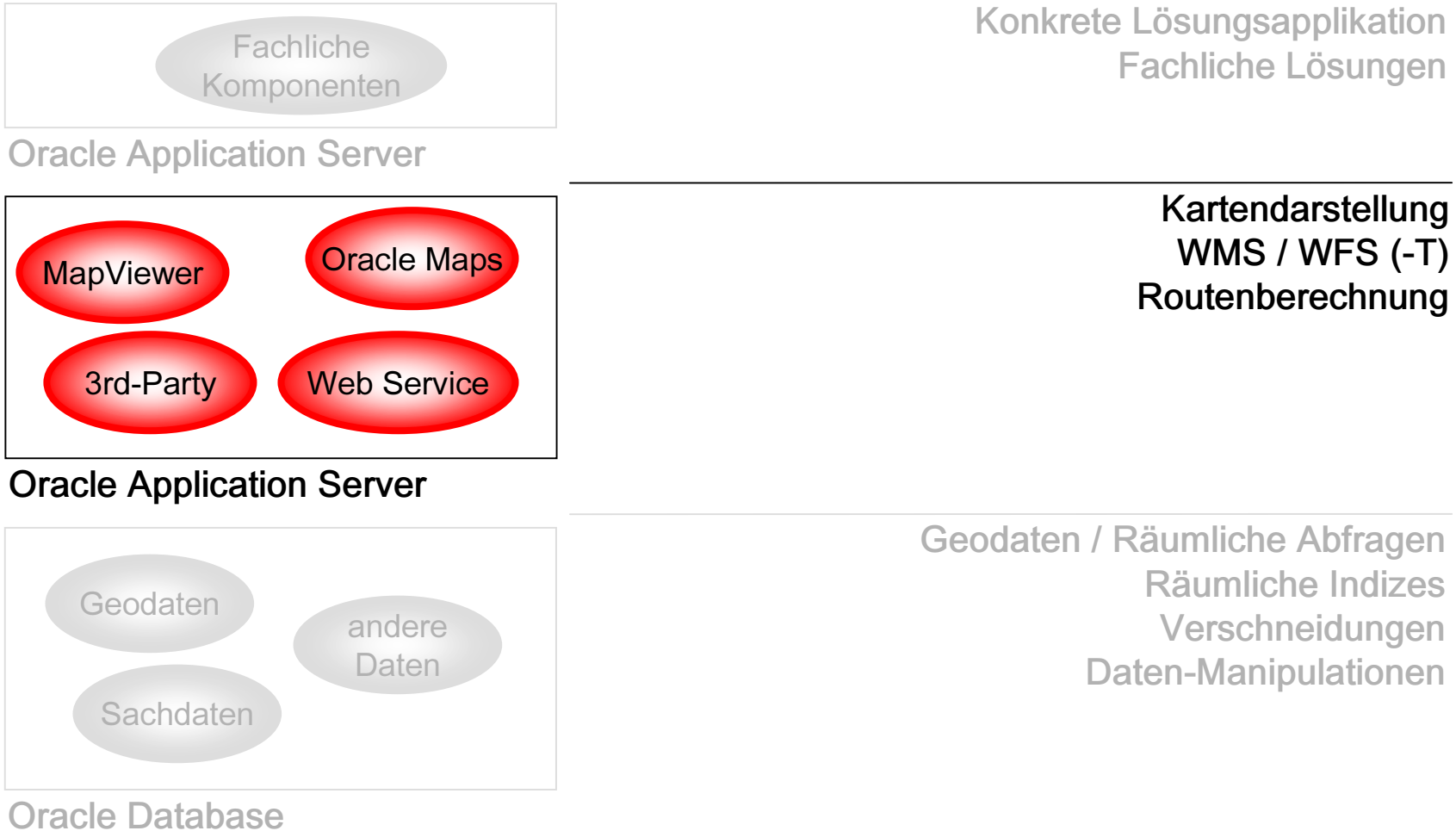



```
select
  ellipsoid_id, ellipsoid_name,
  semi_major_axis, semi_minor_axis
from SDO_ELLIPSOIDS where ellipsoid_id=7030;


ELLIPSOID_ID          7030
ELLIPSOID_NAME        WGS 84
SEMI_MAJOR_AXIS       6378137
SEMI_MINOR_AXIS       -1,896E+09
```

Das Fundament: Die Oracle-Datenbank

Offene Geodaten – offene Plattform!



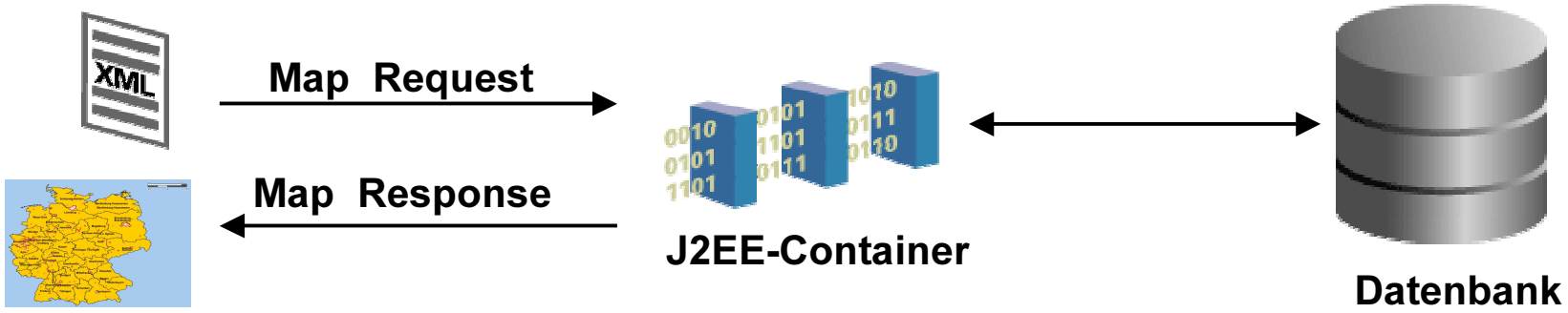
Konkrete Lösungsapplikation
Fachliche Lösungen

Kartendarstellung
WMS / WFS (-T)
Routenberechnung

Geodaten / Räumliche Abfragen
Räumliche Indizes
Verschneidungen
Daten-Manipulationen

Visualisierung der Geodaten
Oracle MapViewer

- Karten-Rendering für ...
  Vektordaten, Rasterdaten, Netzwerke, Topologien
- Unterstützung für WMS (Client / Server)
- Architektur: J2EE
  - Lizenz: Oracle Application Server: Java Edition

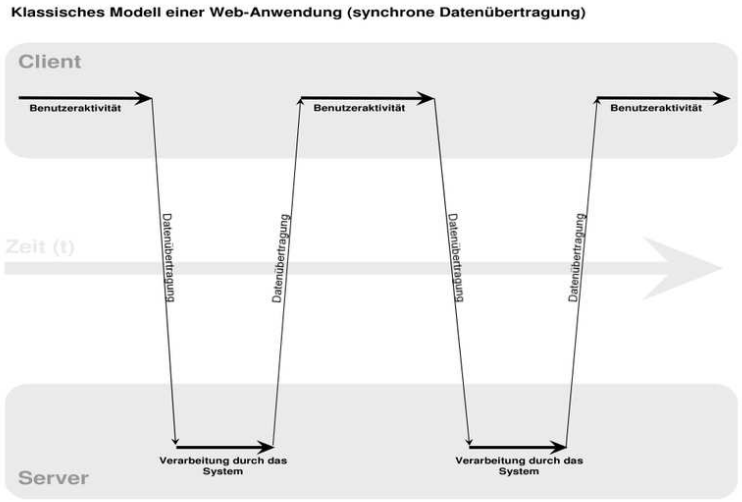


Oracle MapViewer: Oracle Maps!

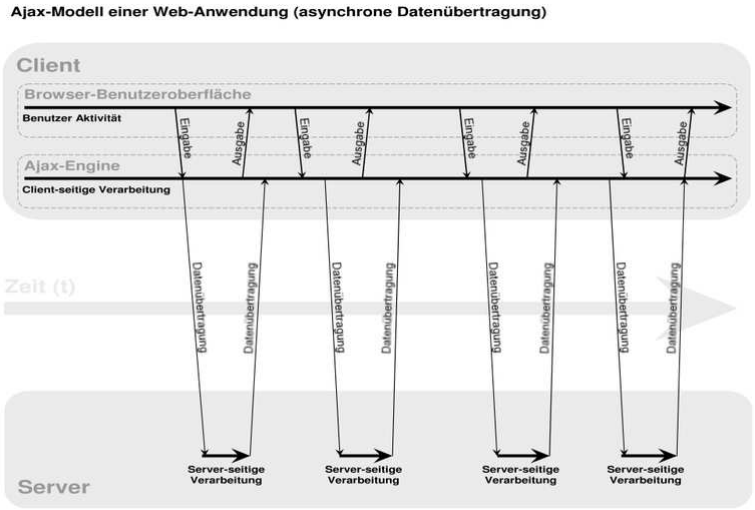
- Kartendarstellung *state-of-the-art*
  - Drag & Drop
  - Einfaches Zoom In & Zoom Out
  - Hinzufügen neuer Geometrien
  - Dynamisches Einblenden weitere Informationen
- AJAX-Technologie
  - JavaScript Client

Event-Modell einer Web-Anwendung

- "AJAX" Event-Modell
  - JavaScript-Komponente sendet *Request* …
    … und empfängt XML *Response*
  - DHTML zur Änderung der Web-Seite



"Klassisches" Modell

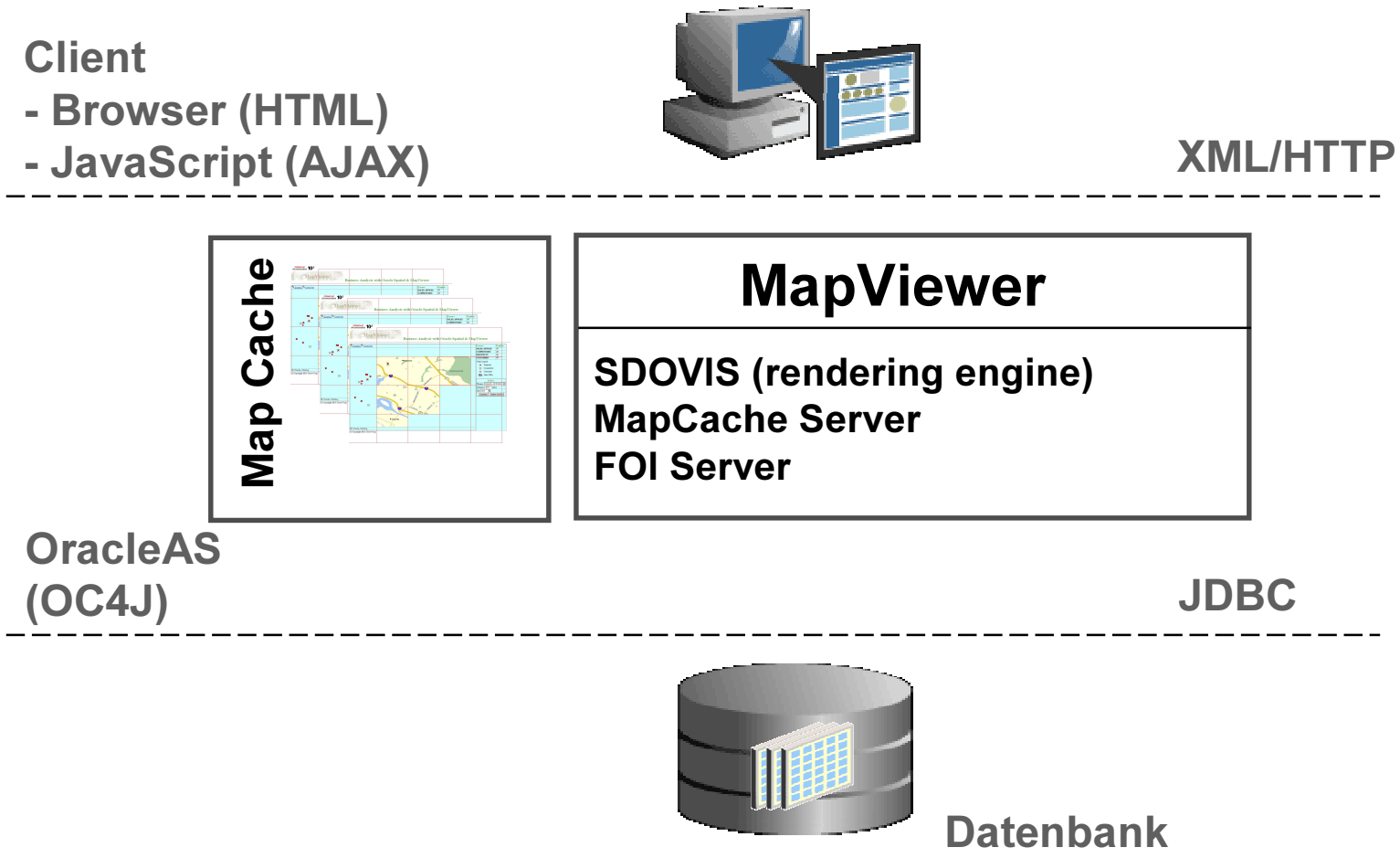


"AJAX"-Modell

Quelle: Wikipedia

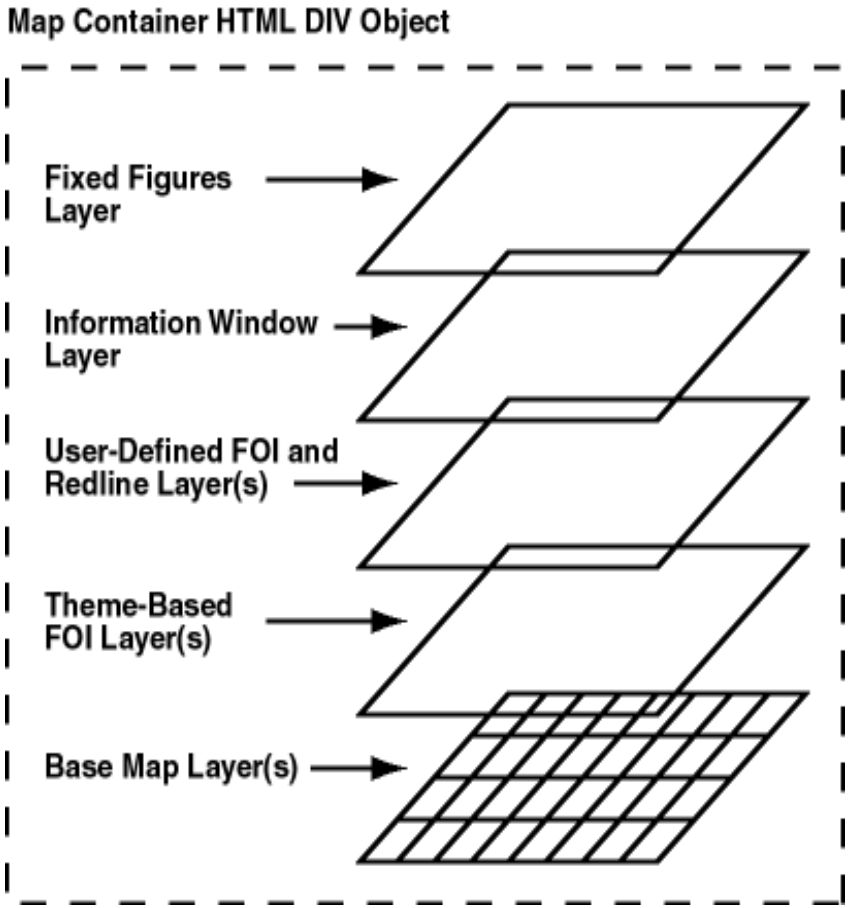
Oracle MapViewer

Oracle Maps: AJAX-Architektur

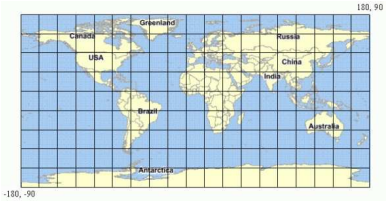


Oracle Maps
Aufbau einer Karte

- *Base map*
  - Statisch
  - Kacheln im Map Cache
- *Theme based FOI*
  - Interaktiv
  - Datenbank-Query
- *User Defined FIO*
  - Interaktiv
  - Anwendungsspezifisch
- *Information Window*



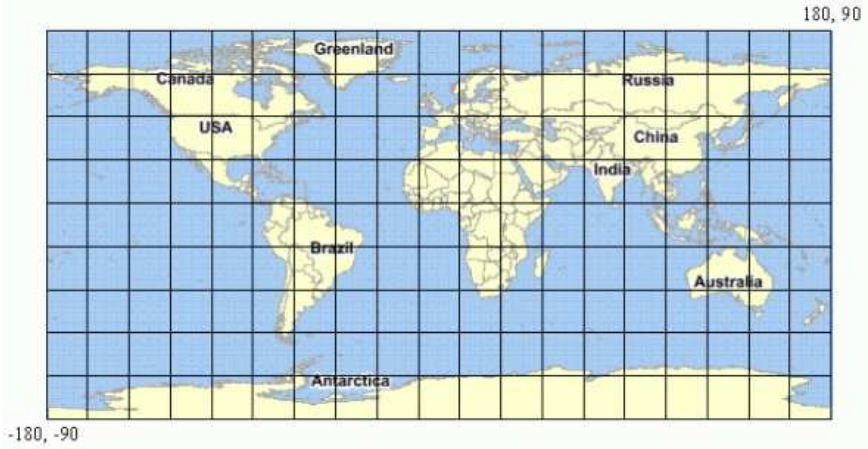
Map Cache Server
Die Basis für schnelle Karten



- Cache für Karten-Bilddateien
  - Kachelung der Karten
  - Caching der einzelnen Kacheln
  - Auslieferung auf Anfrage durch Client API
- Aufbau des Map Cache
  - Nach Zoom Level
  - Base Map
- Speicherung im Dateisystem

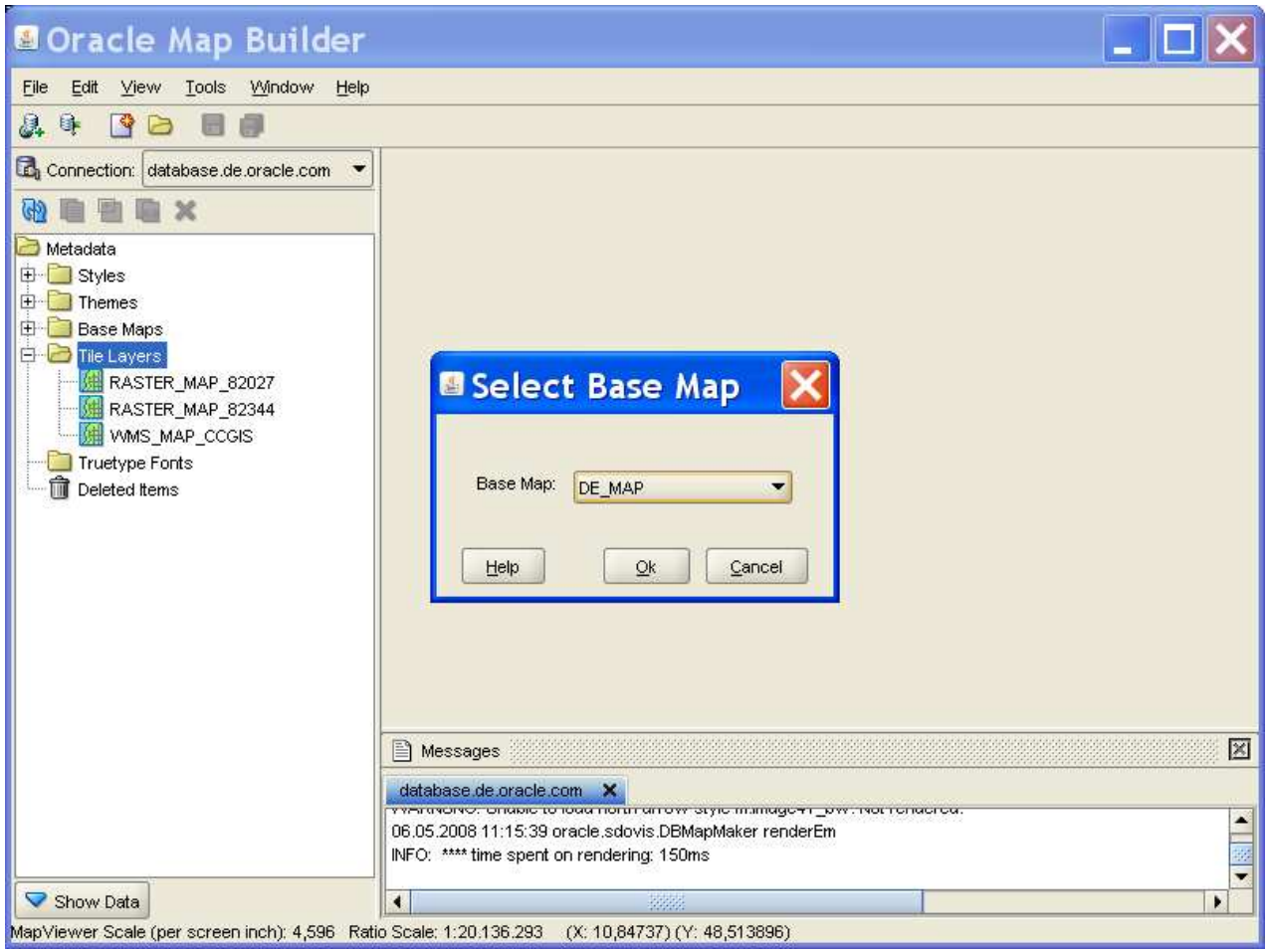
Map Cache und Koordinatensysteme

- Alle Koordinatensysteme möglich
  - Angabe einer SRID in Map Cache Definition
  - Bei Bedarf Transformation
- Geodätische Koordinaten: Alternativen
  - Vorherige Projektion: Bspw. GK 3
  - Kartesische Kachelung



Erzeugen eines MapCache (Tile Layer)

... mit dem MapBuilder



FOI Server

FOI: Feature Of Interest

- Basis: Räumliche SQL-Abfrage
  - Wo wohnen unsere Kunden
  - Welche Vertriebsgebiete machen Umsatz
  - ...
- Abbildung als MapViewer *Theme*
  - Werkzeug: Oracle Map Builder
  - Speicherung in DB: USER_SDO_THEMES
- Ein- und Ausblenden mit JavaScript

AJAX in Oracle Maps

JavaScript Client-API

All Classes
MVBarChartStyle
MVBarInfo
MVBaseMap
MVBucketSeries
MVBucketStyle
MVCircleTool
MVFOI
MVMapDecoration
MVMapView
MVNSDP
MVNumericRangedBucket
MVOverviewMap
MVPieChartStyle
MVPieSlice
MVRectangleTool
MVRedlineTool
MVSdoGeometry
MVSdoPointType
MVStyleColor
MVThemeBasedFOI

Overview Class Tree Index

PREV NEXT FRAMES NO FRAMES

Oracle Maps JavaScript Client API

The Oracle Maps JavaScript client API consists of interfaces and classes that the application can use to access the functionalities provided by the web browser based Oracle Maps Client. MVMapView is the main entry class for all map operations inside the web browser. Together with the rest of the all classes, it provides all essential interfaces for adding logic such as the following to your Web mapping applications:

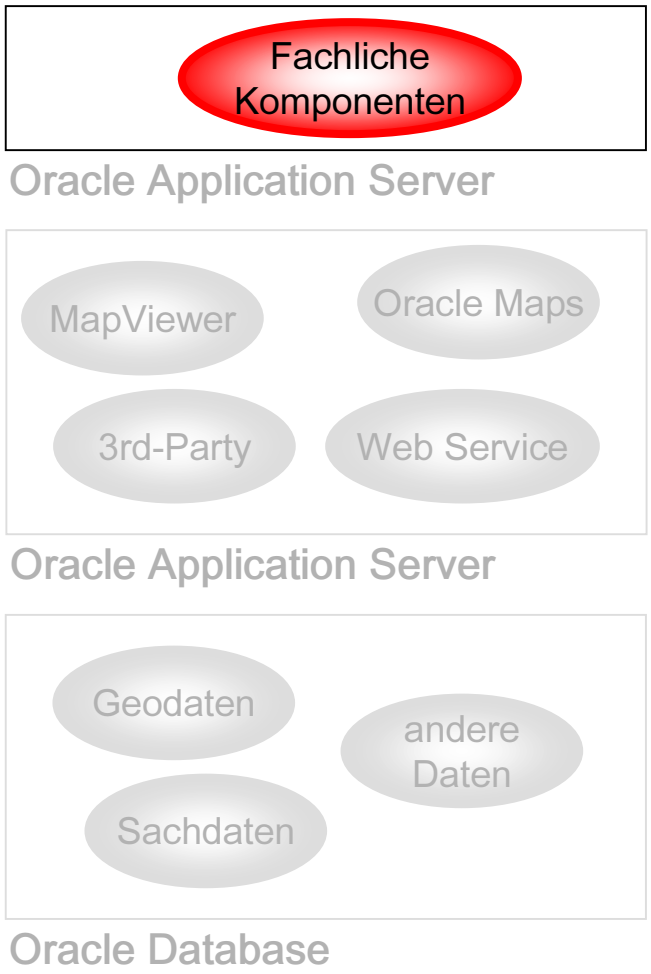
- Create a map client instance and associate it with the map container DIV object created in the Web page.
- Configure map parameters such as map center and map zoom level.
- Create and manipulate base map layers.
- Create and manipulate theme-based FOI layers.
- Create and manipulate user-defined individual FOIs.
- Display an information window on the map.
- Create fixed map decoration, such as map title, custom copyright notes and control buttons.
- Access the built-in utilities such as navigation bar, scale bar, rectangle tool, redline tool, overview map, etc.
- Use event listeners to customize the event handling. You can add event listeners to the MapView, ThemeBasedFOI, and FOI classes using the appropriate API methods.

The following is an example application written with the Oracle Maps JavaScript client API.

```
<html>
<head>
<TITLE>Map Cache Server/Map Client
<script language="Javascript" src="jslib/oraclemaps.js">
```

Das Fundament: Die Oracle-Datenbank

Offene Geodaten – offene Plattform!



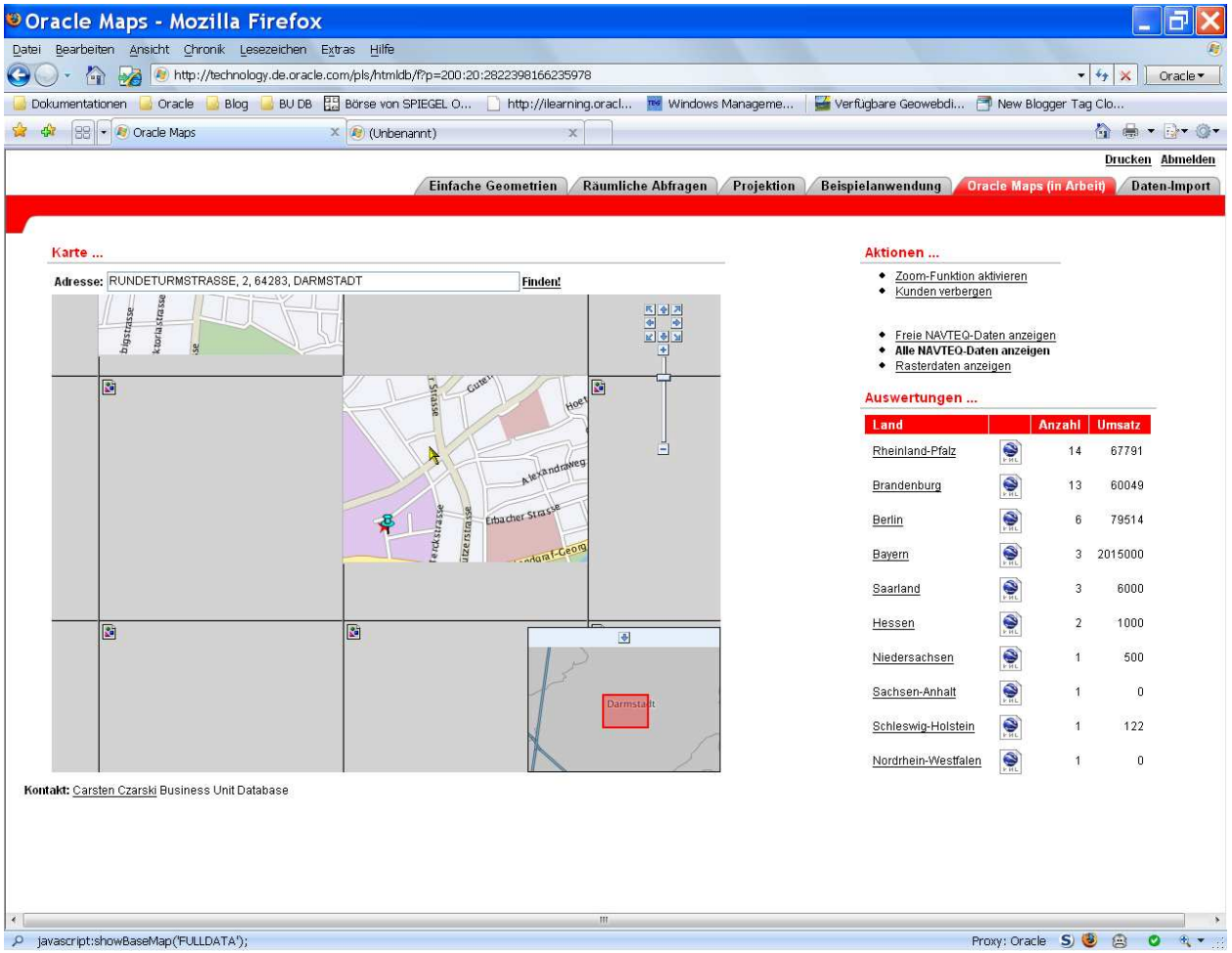
Konkrete Lösungsapplikation
Fachliche Lösungen

Kartendarstellung
WMS / WFS (-T)
Routenberechnung

Geodaten / Räumliche Abfragen
Räumliche Indizes
Verschneidungen
Daten-Manipulationen

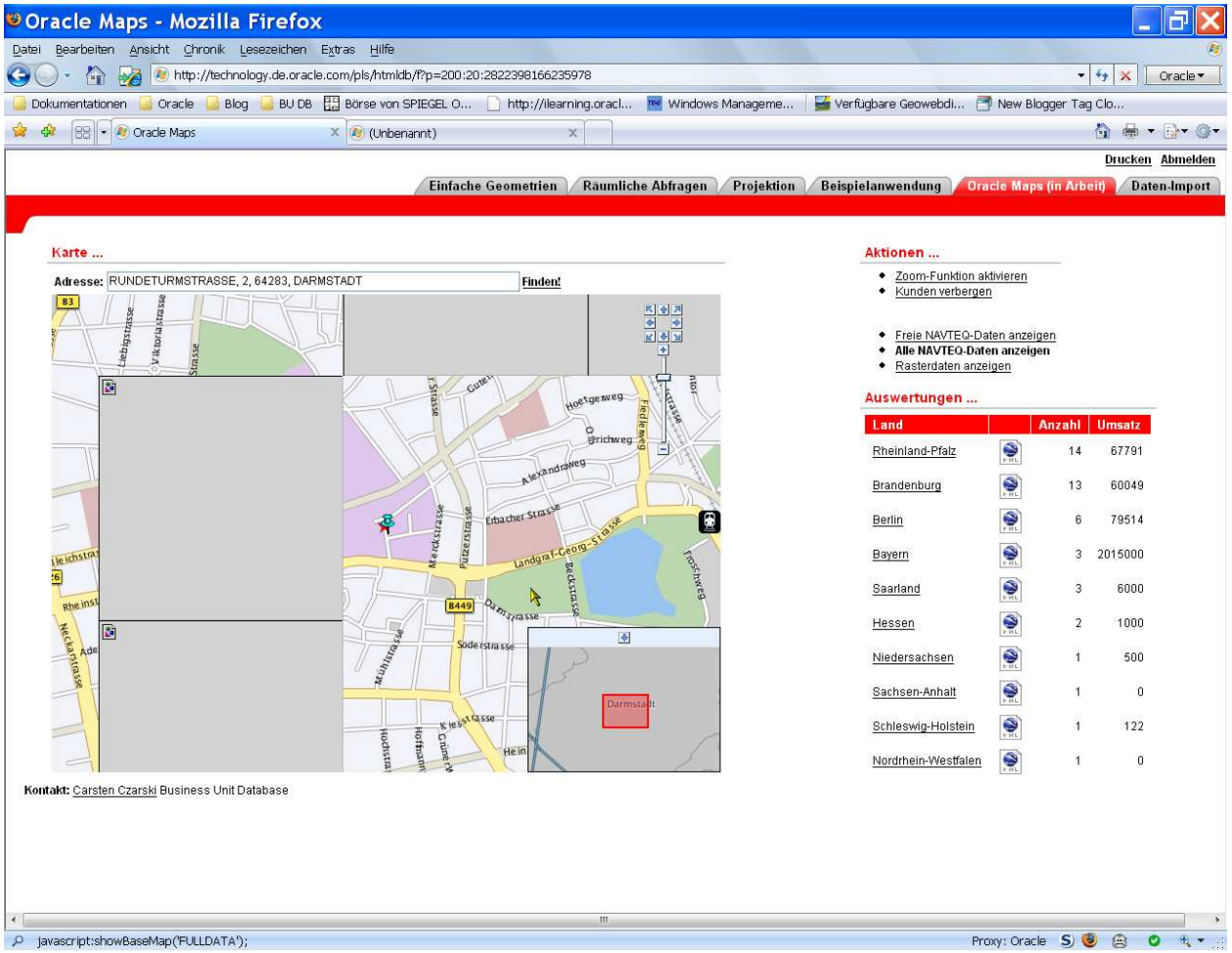
Das Ergebnis:

Eine Anwendung mit Oracle MAPS



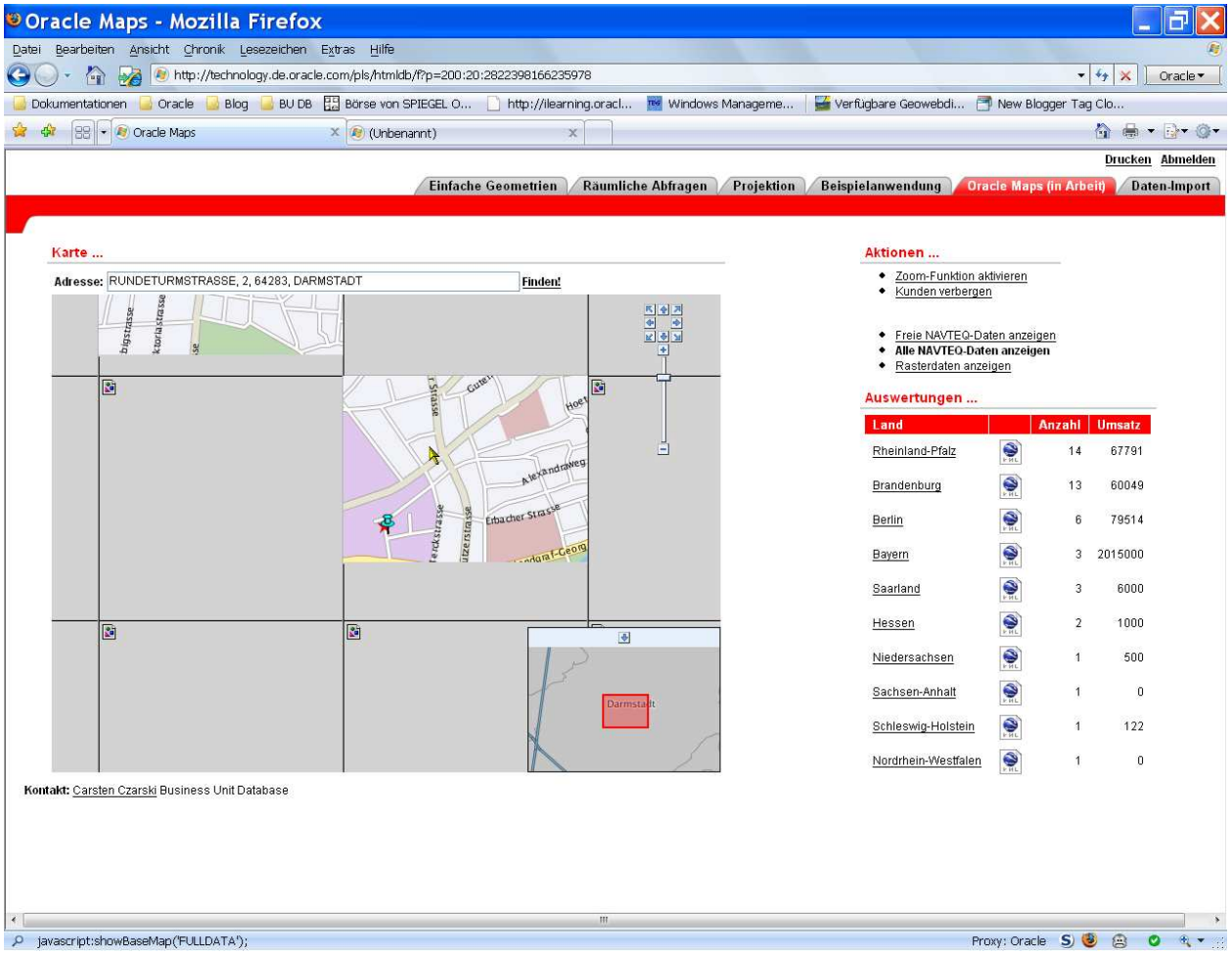
Das Ergebnis:

Eine Anwendung mit Oracle MAPS



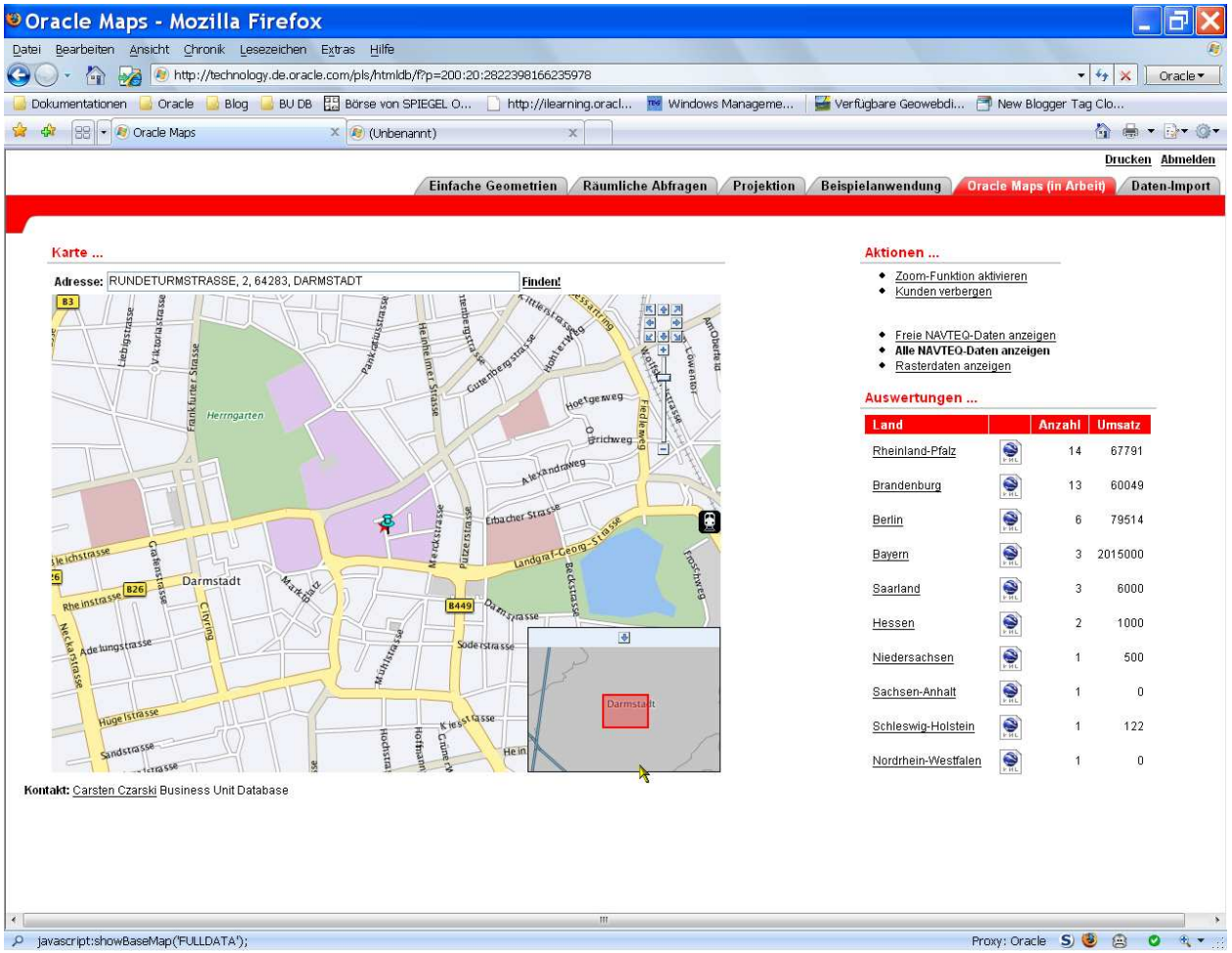
Das Ergebnis:

Eine Anwendung mit Oracle MAPS



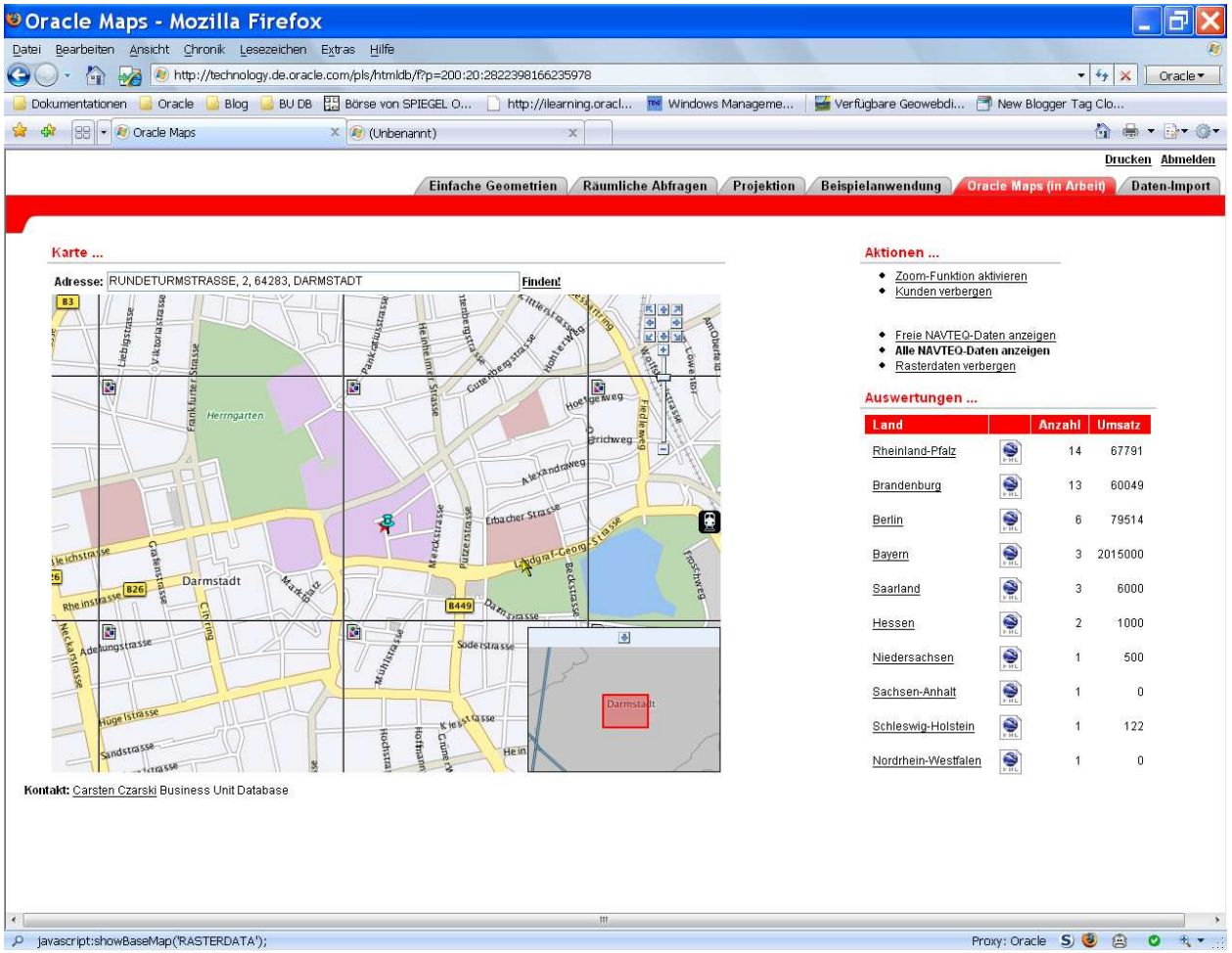
Das Ergebnis:

Eine Anwendung mit Oracle MAPS



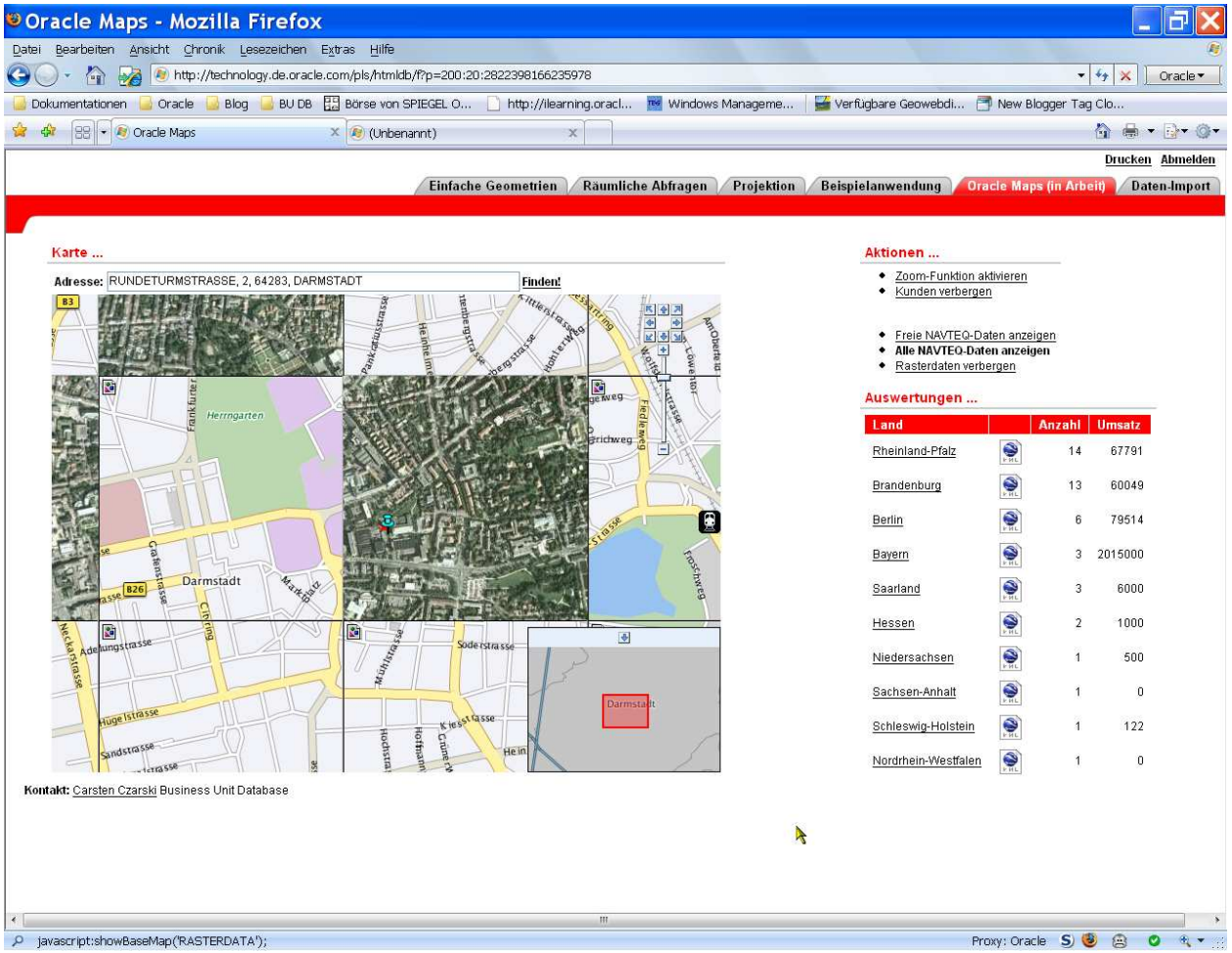
Das Ergebnis:

Eine Anwendung mit Oracle MAPS

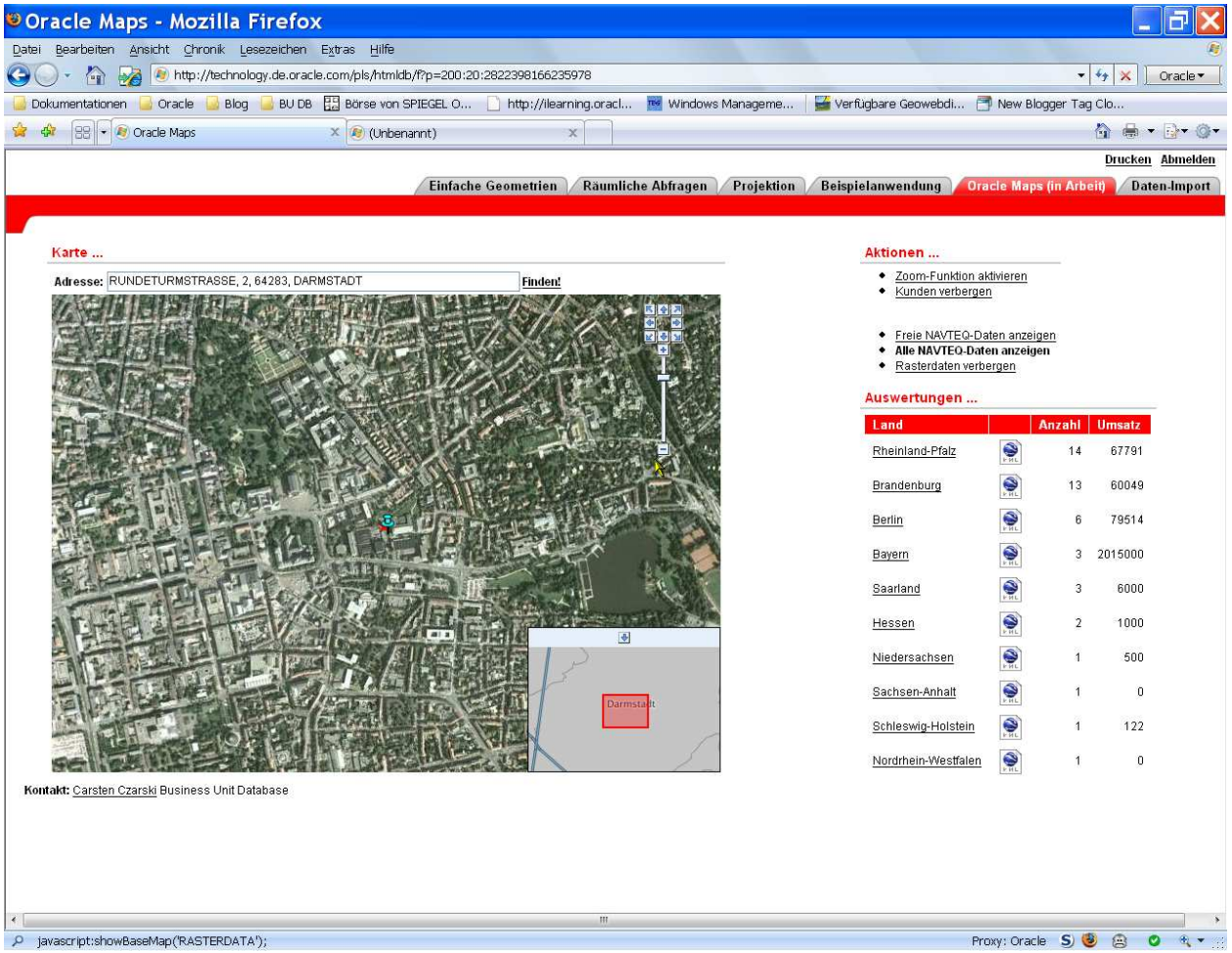


Das Ergebnis:

Eine Anwendung mit Oracle MAPS

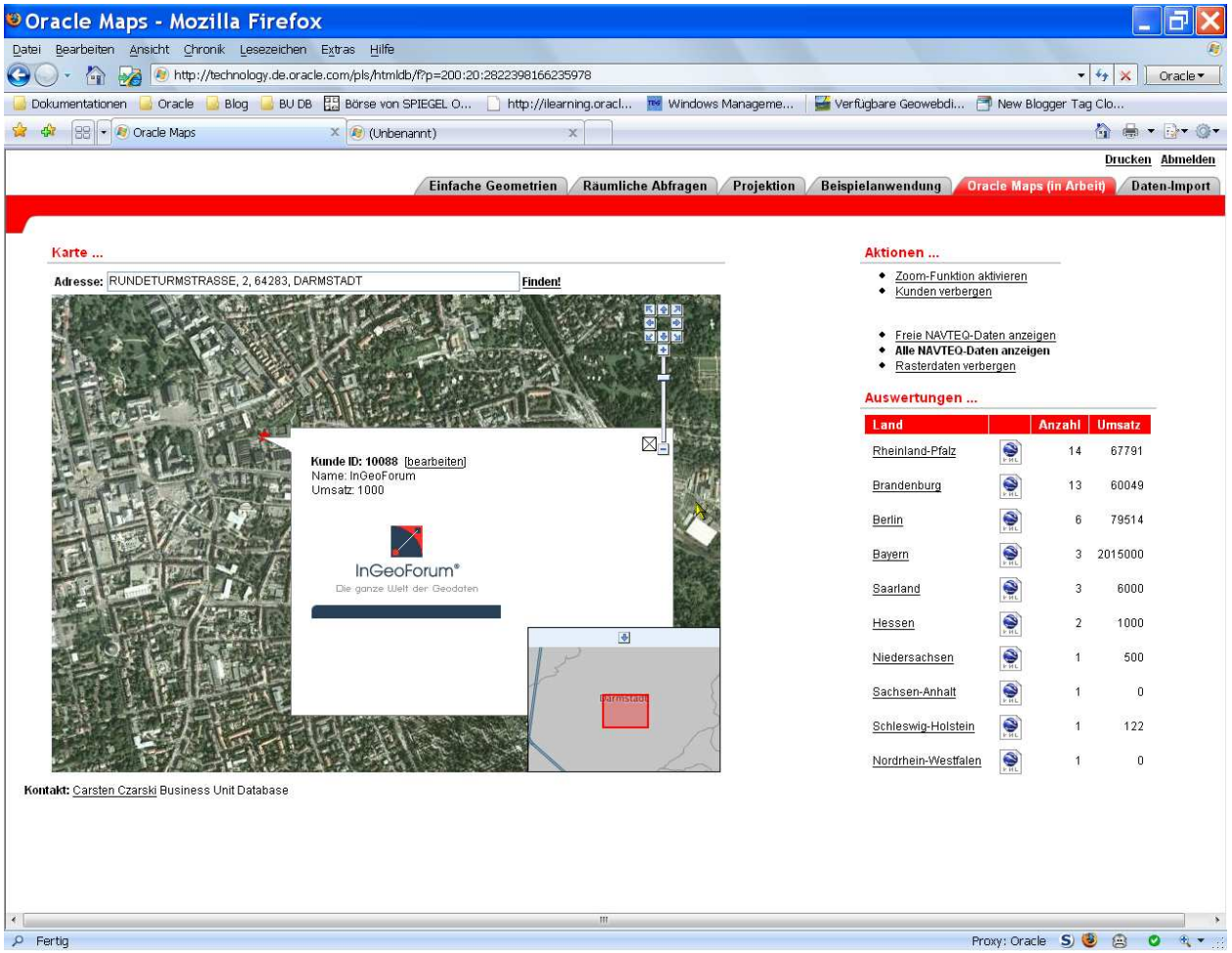


Das Ergebnis:
Eine Anwendung mit Oracle MAPS



Das Ergebnis:

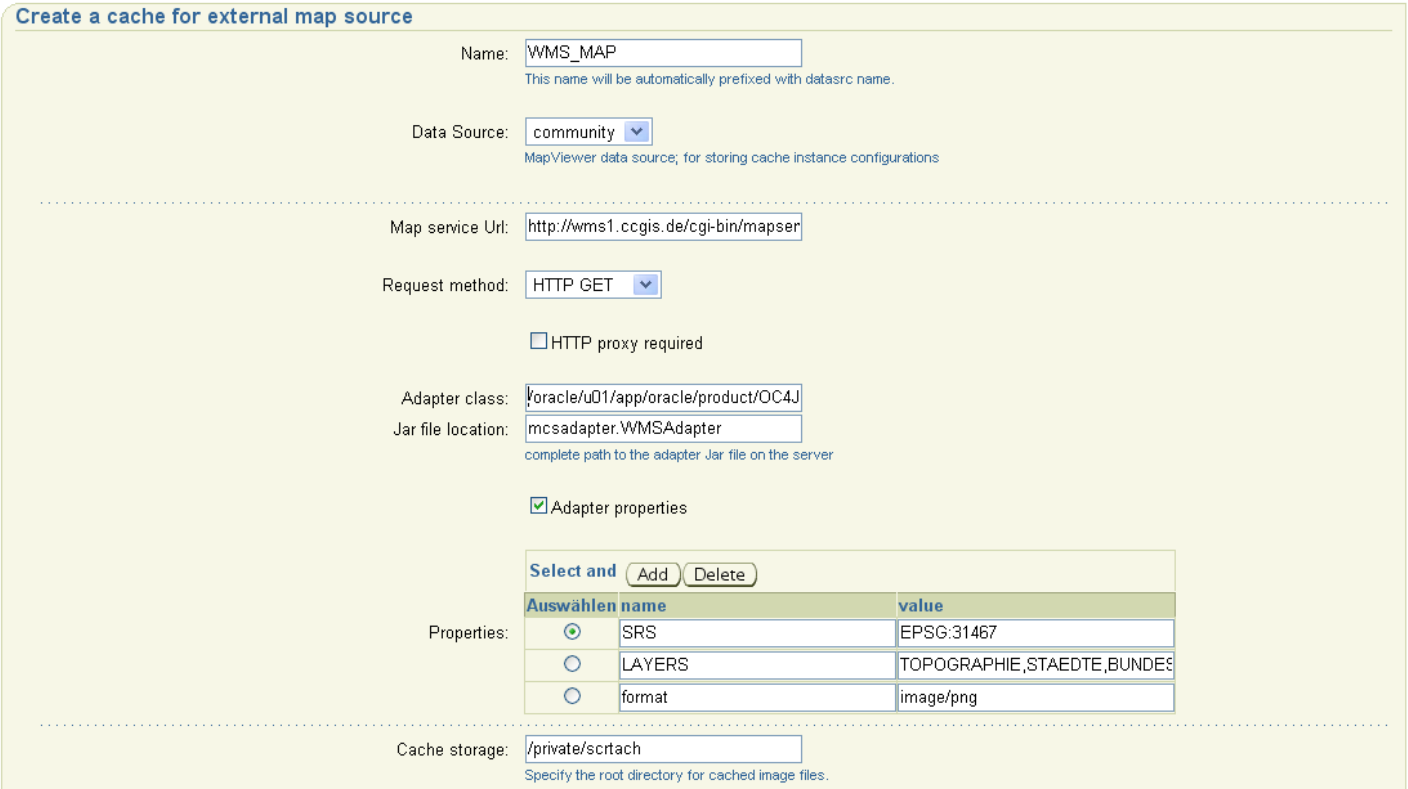
Eine Anwendung mit Oracle MAPS



MAPS weiter denken

Externe Kartenquellen per WMS einbinden

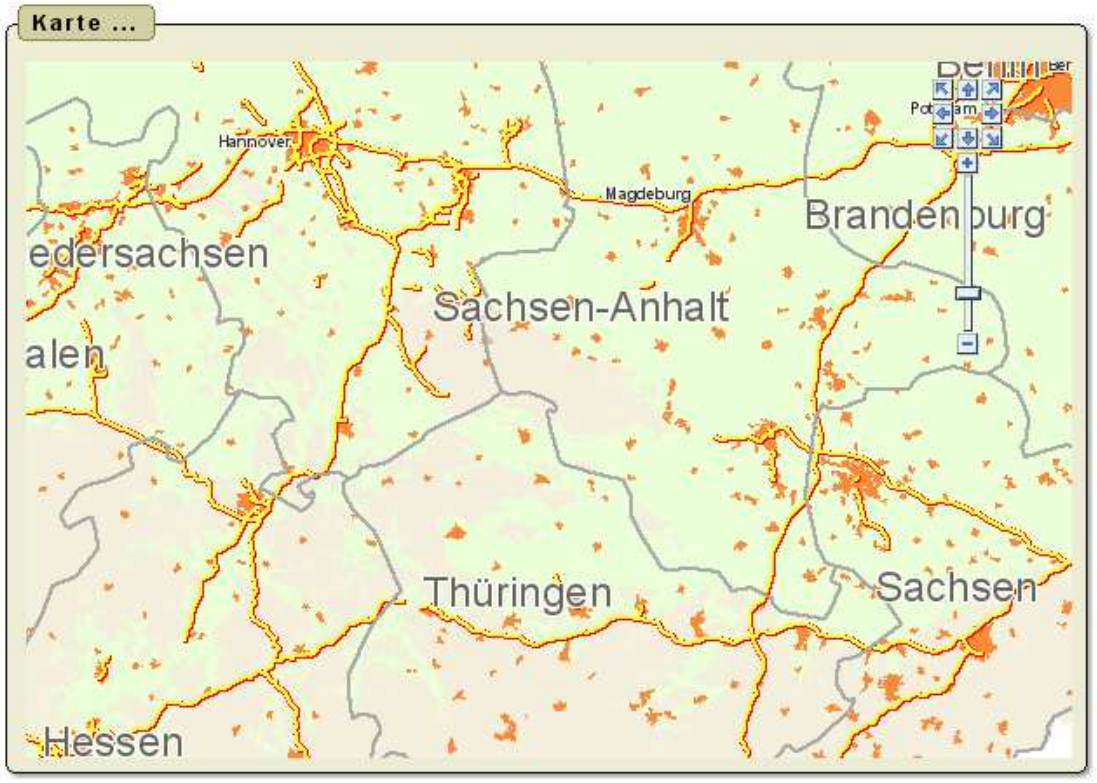
- Interne (Datenbank) und externe Datenquellen
- Externe Datenquelle: WMS



MAPS weiter denken: WMS

Das Ergebnis

- Kacheln werden vom WMS-Server abgerufen und im Map Cache abgelegt ...



QUESTIONS
ANSWERS