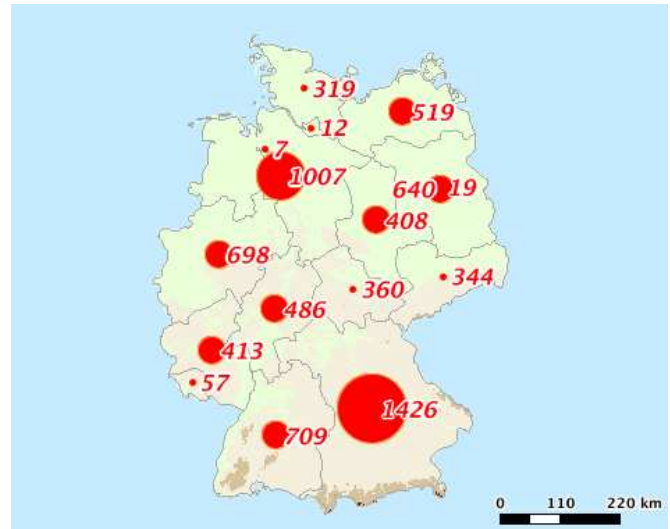


Auf den Ort kommt es an!  
Geodaten und die Oracle-Plattform

*Oracle White Paper*  
*August 2006*

## EINFÜHRUNG: DIE BEDEUTUNG VON GEODATEN

Ein großer Teil geschäftlich relevanter Daten und Informationen ist ortsbezogen. Ob es sich dabei um Adressen von Kunden, um Standorte des eigenen Unternehmens oder um Vertriebs- und Servicegebiete handelt: Eine Lokation ist stets vorhanden.



**Abbildung 1: Praktische Nutzung von Geodaten: Wo sind meine Kunden?**

In den meisten Geschäftsanwendungen wird der geografische Ort jedoch nur indirekt verwendet, etwa als postalische Adresse oder als Kennziffer für das Vertriebsgebiet. Für viele Aufgabenstellungen ist das ausreichend; ein Großteil der operativen Aufgaben wird so abgedeckt. Die Nutzung von exakten Geokoordinaten ist meist Spezialanwendungen im Planungs-, Netzwerk- oder Liegenschaftsbereich vorbehalten, wobei typischerweise spezielle Geo-Informationssysteme (GIS) eingesetzt werden.

Der erfolgreiche Einsatz von Navigationssystemen in vielen Bereichen des täglichen Lebens zeigt deutlich, welchen Stellenwert Anwendungen mit Ortsbezug erreichen können. Auf die gleiche Weise kann die integrierte Verwendung geografischer Informationen in Geschäftsanwendungen deutlich zur Wertschöpfung beitragen.

So wird zum Beispiel die Frage, *welche Kunden im Umkreis um eine bestimmte Filiale wohnen*, durch die Verwendung von Geokoordinaten exakt beantwortet. Solche Informationen können hervorragend als Entscheidungsgrundlage für den Einsatz von Ressourcen oder zur Planung von Kampagnen dienen. Es entstehen vielfältige Möglichkeiten – mit Business Intelligence oder Geomarketing seien hier nur zwei Beispiele genannt.

Den größten Mehrwert entfalten Geodaten jedoch, wenn sie mit Fachdaten kombiniert werden. So lässt sich die Frage nach den Kunden im Umkreis um eine

bestimmte Filiale noch erweitern: *Welche dieser Kunden haben im letzten Jahr das Produkt X gekauft?*

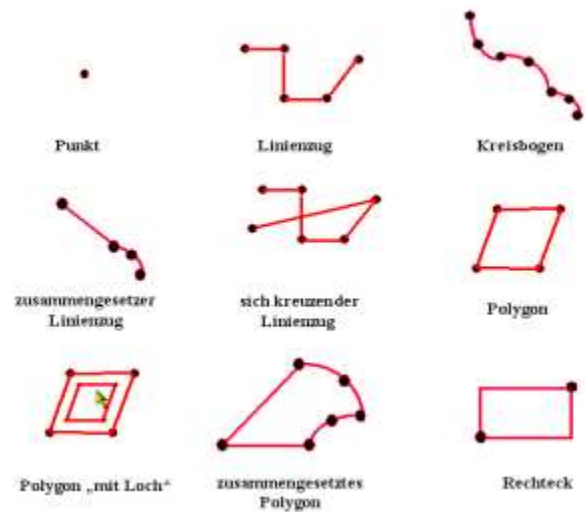
Genau diese mit Fachdaten kombinierte Verwendung von Geodaten wird von der Oracle-Plattform besonders umfassend unterstützt. Das Fundament ist dabei die gemeinsame Speicherung von Geo- und Fachdaten in ein- und derselben Datenbank: Dateninseln gehören der Vergangenheit an.

### ORACLE DATENBANK: BEWÄHRTES FUNDAMENT

Die Oracle-Datenbank hält für Geodaten eigene Datentypen bereit. So werden Vektordaten als *SDO\_GEOMETRY* und Rasterdaten als *SDO\_GEORASTER* gespeichert. Diese Datentypen können wie alle anderen eingebauten Datentypen beliebig in Tabellendefinitionen und PL/SQL-Logik verwendet werden. Die Daten selbst werden zusammen mit den relationalen Daten in den gleichen Speicherbereichen (Tablespaces) gespeichert.

Technische Grundlage dieser Datentypen sind die mit Oracle8 eingeführten objektrelationalen Erweiterungen. So wie der Anwendungsentwickler eigene Datentypen wie *PERSON* oder *KUNDE* definieren kann, ist *SDO\_GEOMETRY* als Objekttyp in der Datenbank definiert – ein *SDO\_GEOMETRY* repräsentiert eine Geometrie. Diese objektrelationale Speicherung von Geodaten entspricht der Spezifikation *SQL with Geometry Types* des Open Geospatial Consortiums.

### SDO\_GEOMETRY: Gespeicherte Koordinaten



**Abbildung 2: Geometrische Primitive in der Oracle-Datenbank**

*SDO\_GEOMETRY* speichert Vektorkoordinaten: Die räumlichen Informationen werden als geometrische Primitive wie Punkte, Linienzüge oder Polygone repräsentiert.

Das Beispiel in Abbildung 3 zeigt, wie die einzelnen Stützpunkte (*vertices*) einer Geometrie mit einem SQL-Kommando abgerufen werden können. Dies macht deutlich: Geodaten sind in der Oracle-Datenbank keine *Black Box*, sondern in einem offenen, standardkonformen und dokumentierten Format gespeichert.

```
select id, x, y
from geodaten e, table(sdo_util.getvertices(e.geometry))
```

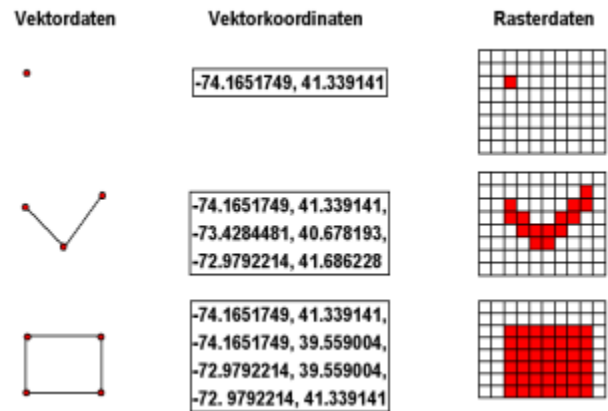
ID	X	Y
1	12.0115001	47.8214
2	11.9985998	47.8229
3	11.9702999	47.8228
:	:	:

**Abbildung 3: Abfrage der einzelnen Koordinaten einer Geometrie**

Diese offene Speicherung der Geodaten ermöglicht, dass nahezu alle am Markt verfügbaren GIS-Werkzeuge mit SDO\_GEOMETRY in der Oracle-Datenbank umgehen können. Werden Geodaten also als SDO\_GEOMETRY gespeichert, können sie mit *verschiedenen* GIS-Werkzeugen und darüber hinaus sogar mit SQL-Kommandos bearbeitet werden. Das Ergebnis ist Unabhängigkeit und hohe Flexibilität bei der Speicherung und Verarbeitung von Geodaten.

### SDO\_RASTER: Verortete Fotos

Neben reinen Vektorkoordinaten treten Geodaten immer häufiger auch als Rasterdaten auf. Beispiele für Rasterdaten sind Satellitenbilder und Orthofotos. Rasterdaten sind insbesondere dann von Vorteil, wenn bei der Visualisierung die Darstellung vieler Details gefragt ist.



**Abbildung 4: Vektordaten vs. Rasterdaten**

Rasterdaten werden in der Oracle-Datenbank als Datentyp SDO\_GEORASTER gespeichert. Der Bezug eines Satellitenbildes oder Orthofotos zur Realwelt wird durch *Georeferenzierung* hergestellt. Die sich daraus ergebenden Koordinaten für die Ecken der *Rasterkachel* können wiederum mit räumlichen Operatoren abgefragt werden. So kann die Datenbank nicht nur die Frage nach den Kunden im Umkreis

um eine Filiale beantworten, sondern zusätzlich die passenden Luftbilder ausliefern.

### Nutzung der Geodaten: Räumliche Abfragen mit SQL

Mit der Verwendung von SDO\_GEOMETRY oder SDO\_GEORASTER werden Daten in der Datenbank zu Geoinformationen. Räumliche Abfragen (*SDO\_RELATE*) oder geometrische Operationen wie Schnittmengenbildung (*SDO\_INTERSECTION*) sind nur zwei Beispiele für die umfassende Funktionalität, die nun bereitsteht.

```
select l1.name
from laender l1, laender l2
where l2.name = 'NRW'
and SDO_RELATE(
  l1.grenze, l2.grenze,
  'mask=touch querytype=window'
) = 'TRUE'
```



Abbildung 5: Räumliche Abfrage mit SQL

Abbildung 5 zeigt ein Beispiel für eine räumliche Abfrage. Mit der SQL-Funktion SDO\_RELATE wird ermittelt, welche Bundesländer direkt an das Land „NRW“ grenzen (*MASK=TOUCH*). Diese SQL-Abfrage kann nun beliebig mit anderen relationalen Fachdaten kombiniert werden. Neben dem Test auf direkte Berührung kennt die Oracle-Datenbank natürlich noch eine Reihe weiterer räumlicher Abfragen und Funktionen. Einige Beispiele dafür sind:

- **SDO\_INSIDE:**  
Befindet sich eine Geometrie komplett innerhalb einer anderen?
- **SDO\_NN:**  
Welche sind die nächsten Nachbarn zu einer Geometrie?
- **SDO\_WITHIN\_DISTANCE:**  
Welche anderen Geometrien befinden sich innerhalb einer bestimmten Distanz zu einer Geometrie?
- **SDO\_GEOM.SDO\_CENTROID:**  
Wo ist der Schwerpunkt der Geometrie?
- **SDO\_GEOM.SDO\_INTERSECTION:**  
Welches ist die „Schnittmenge“ zweier Geometrien?

Damit räumliche Abfragen auch bei großen Datenmengen performant arbeiten, ist die Indizierung der Daten unabdingbar. Die Oracle-Datenbank bietet speziell für Geodaten den *R-Tree-Index* an. Dieser wird mit dem für den Datenbank-

Administrator vertrauten Kommando `CREATE INDEX` angelegt und bei räumlichen Abfragen automatisch herangezogen.

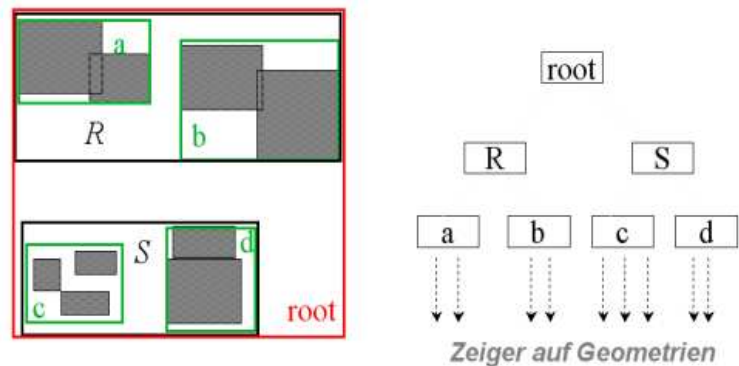


Abbildung 6: Aufbau eines R-Tree-Indexes

### Koordinatensysteme: Bezug zur Realität

Welchen tatsächlichen Ort auf der Erdoberfläche eine Geokoordinate beschreibt, hängt vom verwendeten Koordinatensystem ab. Es besagt, wie die einzelnen Koordinaten zu interpretieren sind. Zum Beispiel muss man zur Koordinate „10, 50“ wissen, dass Längen- und Breitengrade gemeint sind. Die Oracle-Datenbank unterstützt insgesamt über 1000 Koordinatensysteme:

- *Geodätische Koordinatensysteme* arbeiten mit Längen- und Breitengraden. Nur mit geodätischen Koordinaten können Distanzberechnungen exakt durchgeführt werden, da sie berücksichtigen, dass die Erde eine Kugel (oder genauer: ein Spheroid) ist. Mit typischen Besonderheiten wie Distanzberechnungen über die Pole hinweg kann die Datenbank problemlos umgehen.
- *Projizierte Koordinatensysteme* werden auf jeder Straßenkarte verwendet. Mit mathematischen Verfahren (z.B. *Gauß-Krüger*) werden die Koordinaten auf eine ebene Fläche projiziert. Eine Koordinate in einem projizierten Koordinatensystem entspricht stets einem Ort auf der Erdoberfläche.
- *„Non-Earth“-Koordinatensysteme* werden für Geometrien verwendet, welche keinen Bezug zu Lokationen auf der Erdoberfläche haben.

Geometrien können zwischen den verschiedenen Koordinatensystemen transformiert werden. So ist es eine Frage der Mathematik, eine Geometrie vom Koordinatensystem „WGS84“ (Längen-/Breitengrade) in ein projiziertes Koordinatensystem umzurechnen. Und gerade hier kann die Oracle-Datenbank ihre Vorteile unter Beweis stellen. Geht es darum, ganze Datenbestände zwischen verschiedenen Koordinatensystemen zu transformieren, so erledigt die Datenbank diese Aufgabe direkt auf den Geodaten – aufwändige Transfers zwischen der Datenbank und einer Spatial-Middleware entfallen.

## Von der Postadresse zur Koordinate: Geocoding

Wie bereits am Anfang des Dokuments beschrieben, ist ein großer Teil der geschäftlich relevanten Daten und Informationen ortsbezogen. In den meisten Fällen liegt jedoch nur die postalische Adresse vor, so dass sich die Frage stellt, wie man überhaupt zur Geokoordinate kommt.

**Riesstr. 25  
80992 München**

**Länge: 11,5367  
Breite: 48,1800**



**Abbildung 7: Von der Adresse zur Koordinate: Geocoding**

Die Ermittlung einer Geokoordinate aus einer postalischen Adresse wird als *Geocoding* bezeichnet. Zu beachten ist dabei, dass postalische Adressen vielfach Fehler enthalten oder unvollständig sind. Ein Geocoding-Algorithmus muss dies berücksichtigen und bei Bedarf entsprechend fehlertolerant arbeiten.

Seit Oracle10g kann Geocoding direkt in der Datenbank durchgeführt werden. Die Geocoding-Software wurde gemeinsam mit NAVTEQ entwickelt, ist in die Datenbank integriert und mit der Spatial Option verfügbar. Für das Geocoding selbst ist wiederum ein Geodaten-Bestand nötig – dieser kann von Oracle-Partnern wie NAVTEQ direkt im passenden Format erworben werden.

## Geodaten in der Datenbank: Weitere Aspekte

Die Vorteile der integrierten Speicherung von Geo- und Fachdaten enden nicht bei kombinierten Abfragen. Die enge Integration in ein- und derselben Datenbank bietet weitere Vorteile. Einige Beispiele dafür sind:

- Mit Hilfe von *Constraints* und *Triggern* kann die Integrität zwischen Geodaten und fachlichen Daten sichergestellt werden. Die Datenbasis der Anwendung ist vollständig – aufwändige Implementierungen zur Sicherstellung der Integrität fallen weg.
- Zugriffsrechte können mit Datenbankmitteln wie der *Virtual Private Database* direkt in der Datenbank hinterlegt werden. So können sensible Geodaten effizient geschützt werden. Die Datenbank setzt die Zugriffsrechte unabhängig von der Anwendung durch.
- Bestehende Betriebskonzepte (Backup & Recovery) können auch für die Geodaten genutzt werden.

## GEODATEN FÜR DEN ENDANWENDER: VISUALISIERUNG

Nahezu jede mit Geodaten arbeitende Anwendung visualisiert dem Endbenutzer die Daten als Karte. Ein typisches Beispiel für eine solche Visualisierung ist Abbildung 1 im ersten Abschnitt dieses Dokumentes.

Zur Erzeugung solcher Karten bietet sich der Oracle *MapView* an. Der MapViewer, mit dem Oracle Application Server (Java Edition) verfügbar, ist eine hochperformante Software-Komponente zur Erstellung von Karten anhand in der Oracle-Datenbank gespeicherter Geodaten. Sowohl Vektordaten als auch Rasterdaten werden komplett unterstützt.

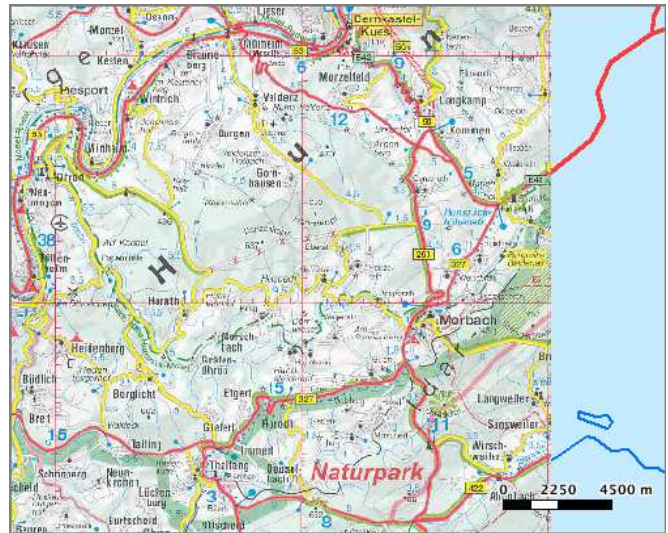


Abbildung 8: Darstellung von Rasterdaten mit dem MapViewer

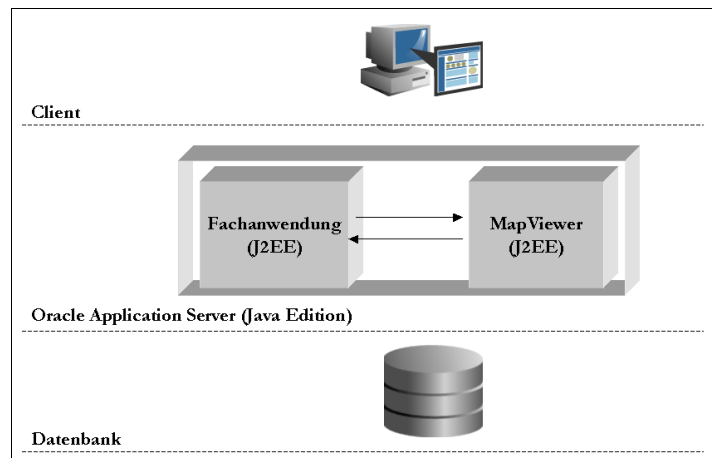
Sind nicht alle Daten zur Erstellung der Landkarte in der Datenbank verfügbar, so bietet der MapViewer die Möglichkeit, mithilfe von *Web Map Service (WMS)* Kartendaten von einem entfernten Server abzurufen. Diese Kartendaten werden in die endgültige Landkarte integriert. Natürlich kann der MapViewer auch selbst als WMS-Server auftreten und seine Karten so für andere Server bereitstellen.

Die Karten können als einfaches Bild (PNG, JPEG, GIF und andere) oder auch als SVG (*Scalable Vector Graphics*) erstellt werden. Der besondere Vorteil von SVG ist die Möglichkeit, Karten *aktiv* zu gestalten – wenn der Anwender etwas in der Karte anklickt, wird ein *JavaScript-Callback* ausgelöst und die Anwendung kann weitere Informationen anzeigen.

### Oracle MapViewer: Offene, standardkonforme Architektur

Technisch präsentiert sich der MapViewer als J2EE-Anwendung, welche in die *Oracle Containers for J2EE (OC4J)* installiert wird (Abbildung 9). Über offene Standards wie XML und HTTP können Dritt-Anwendungen mit dem MapViewer kommunizieren, Anfragen (*Map Requests*) stellen und fertig zusammengestellte Karten (*Map Responses*) empfangen.

Dank dieser offenen Architektur lässt sich der MapViewer in jeder IT-Landschaft und von jeder Applikation aus nutzen. Die Einbindung in Client/Server-Anwendungen (*Rich Clients*) ist ebenso denkbar wie die Nutzung in einer Web-Architektur.



**Abbildung 9: Architektur des Oracle MapViewer**

Ogleich der MapViewer als J2EE-Anwendung implementiert ist, ist seine Nutzung nicht auf die Java-Welt beschränkt. Da die Kommunikation mit Hilfe von XML über das HTTP-Protokoll erfolgt, kann jede Entwicklungs- und Laufzeitumgebung den MapViewer nutzen. Für Java und PL/SQL werden bereits fertige Client-Bibliotheken mitgeliefert.

### Karten definieren mit dem Oracle MapBuilder

Grundlage einer vom MapViewer erstellten Karte sind zuvor gemachte Definitionen über ihr Aussehen (*Styles, Themes und Maps*).

- Der MapViewer *Style* legt fest, wie etwas gezeichnet werden soll. Ein Style könnte also definiert sein als: Rote Linie, 2 Punkt dick.
- Ein MapViewer *Theme* ist eine Sammlung von Geodaten, welche mit einem Style verknüpft werden. Ein Theme könnte also wie folgt definiert sein:  
Autobahnen werden als Rote Linie, 2 Punkt gezeichnet. Die Daten liegen in der Tabelle AUTOBAHN in der Spalte GEOMETRY.
- Eine Karte (MapViewer *Map*) ist eine Zusammenstellung von Themes, welche beim Erstellen wie Folien übereinandergelegt werden.

Zur Definition der Styles, Themes und Maps kommen wiederum offene Standards wie XML und SVG zum Einsatz.

Mit dem *Oracle MapBuilder* können diese Definitionen ohne Programmier- oder Datenbankkenntnisse hinterlegt werden. Auch Fachanwender können somit das Aussehen der generierten Karten beeinflussen.

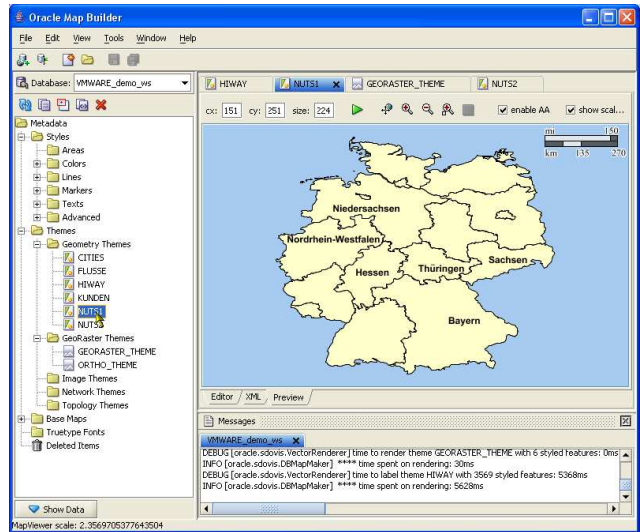


Abbildung 10: Karten definieren mit dem MapBuilder

## ZUSAMMENFASSUNG

Mit der Oracle-Datenbank und dem MapViewer im Oracle Application Server stellt Oracle leistungsfähige Technologie zur Verarbeitung und Darstellung von Geoinformationen bereit. Größtmögliche Offenheit ist dabei eines der wichtigsten Architekturkriterien.

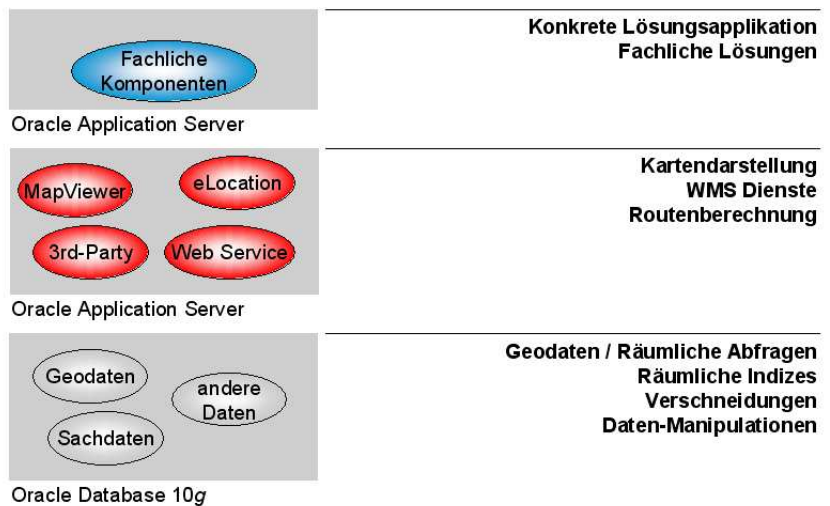


Abbildung 11: Offene Geodaten - Offene Plattform

Die Speicherung der Geodaten in der Oracle-Datenbank ist ein stabiles Fundament für jede Applikation mit räumlichen Funktionalitäten. Das Format (SDO\_GEOMETRY und SDO\_GEORASTER) ist offen, dokumentiert und es entspricht den Standards des Open Geospatial Consortiums. Nahezu alle Anbieter von GIS-Software unterstützen die Speicherung von Geodaten in der Oracle-Datenbank.

Wenn es um die Visualisierung geht, ist der Oracle MapViewer eine leistungsfähige, skalierbare und kostengünstige Alternative. Die Implementierung als J2EE-Anwendung und die Kommunikation über XML und HTTP machen die Anwendungsentwicklung und die Integration in bestehende IT-Landschaften einfach.

Somit bietet es sich an, die Geodaten-bezogenen Aufgaben einer Geschäftsanwendung durch die Technologie der Oracle-Plattform erledigen zu lassen; sie bietet mächtige räumliche Funktionalität auf der einen und Offenheit, Flexibilität und damit Investitionsschutz auf der anderen Seite.

#### **WEITERE INFORMATIONEN**

- [Oracle Technology Network: Location Services](#)
- [Beispieldaten für das Geocoding von NAVTEQ](#)
- [White Paper: Oracle Locator and Oracle Spatial Option](#)



Auf den Ort kommt es an! Geodaten und die Oracle-Plattform  
August 2006  
Author: Carsten Czarski  
Contributing Authors: Wolfgang Wagner

ORACLE Deutschland GmbH  
Riesstr. 25  
80992 München  
Deutschland

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[www.oracle.com](http://www.oracle.com)

Copyright © 2006, Oracle. All rights reserved.  
This document is provided for information purposes only  
and the contents hereof are subject to change without notice.  
This document is not warranted to be error-free, nor subject to  
any other warranties or conditions, whether expressed orally  
or implied in law, including implied warranties and conditions of  
merchantability or fitness for a particular purpose. We specifically  
disclaim any liability with respect to this document and no  
contractual obligations are formed either directly or indirectly  
by this document. This document may not be reproduced or  
transmitted in any form or by any means, electronic or mechanical,  
for any purpose, without our prior written permission.  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective owners.