OracleMaps mit Microsoft ASP.NET nutzen

Autor: Arnd Spiering, g.on experience gmbh

Oracle Maps mit Microsoft ASP.NET verbinden, das bedeutet eine Brücke zwischen 2 unterschiedlichen IT Welten zu schaffen. Viele Unternehmen setzen ASP.NET als Plattform für Webanwendungen ein. Um diese Plattform mit einer sehr leistungsstarken und offenen räumlichen Komponente zu erweitern, bietet Oracle mit der Produktpalette um Oracle Spatial und Microsoft .NET alle Möglichkeiten. Dieser Artikel richtet sich an Entwickler und Administratoren, die Anwendungen betreiben bzw. entwickeln, die auf beiden Technologien aufsetzen. Es wird deutlich, welches Potential durch die räumlichen Funktionen aus Oracle Spatial durch Oracle Maps in Webanwendungen zur Verfügung gestellt wird, und wie einfach dieses mit ASP.NET Anwendungen kombiniert werden kann.

Oracle Spatial und Oracle Locator bieten räumliche Funktionen und Speicherstrukturen für Oracle Datenbanken. Oracle Locator bietet dabei die Basisfunktionen, Oracle Spatial als Option für die Enterprise Edition die erweiterte Funktionalität für GIS Anwendungen und LocationBasedServices.
Oracle MapViewer ist eine Visualisierungskomponente, mit der Karten bzw. Darstellungen aus graphischen Daten erzeugt werden. Oracle Maps ist eine Ajax GUI, die im Hintergrund über MapViewer Komponenten Karten erzeugt.
Microsoft .NET wird seitens Oracle mit ODP.NET unterstützt.
In diesem Artikel wird eine Oracle 10g R2 Datenbank mit der Spatial Option eingesetzt.
Dieser Artikel zeigt Installationen, Konfigurationen und ein Beispielprojekt in Microsoft Visual Studio. Eine eigene Ajax Implementierung erweitert die Karteninhalte.

Microsoft ASP.NET

Um Microsoft ASP.NET nutzen zu können, ist ein Windows Betriebssystem mit einem Internet Information Server (IIS) erforderlich. Das ist z.B. in Windows XP Professional oder allen Serverbetriebssystemen enthalten.
Der IIS lässt sich als Zusatzkomponente zum Betriebssystem installieren.

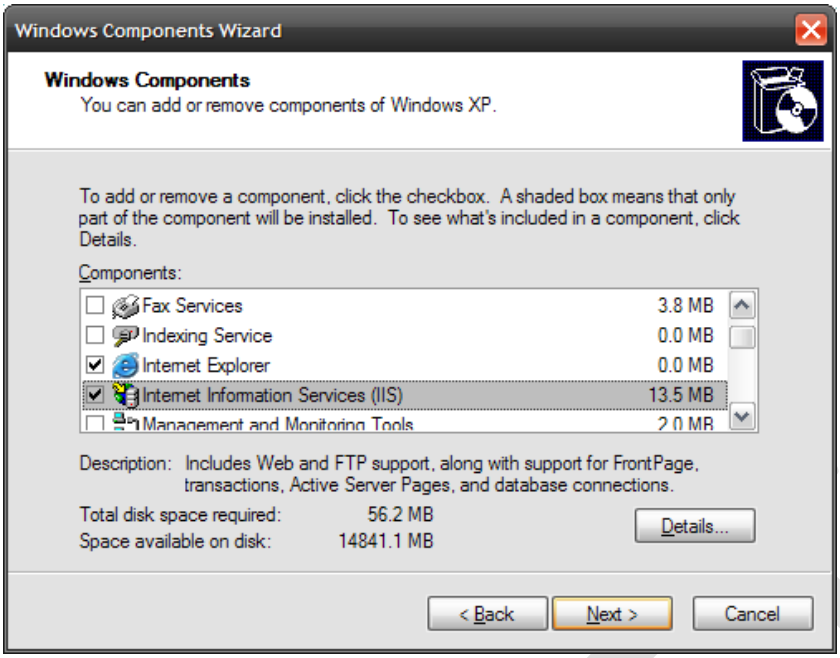

Abbildung 1: Windows Komponenten verwalten

Vor der IIS Installation muss das .NET Framework 2.0 installiert sein. Ist der IIS in einer Standardinstallation enthalten, kann über http://localhost die Startseite des IIS aufgerufen werden. Der Internet Information Service wird als Snap In für die Computer Management Konsole angeboten.

Abbildung 2: IIS Startseite im Hintergrund, im Vordergrund das IIS Snap In

Es werden neue Windows Dienste installiert, die wichtigsten sind *IIS Admin* und der *World Wide Web Publishing* Dienst. Sie sollten beachten, dass der Computer bei einer Verbindung zum Internet jetzt als Webserver funktioniert, falls der Port 80 nicht gesperrt ist.

Oracle Spatial

Ob Oracle Spatial installiert ist, lässt sich mit folgender SQL Abfrageüberprüft werden:

```
select COMP_NAME, STATUS from DBA_REGISTRY where COMP_NAME
='Spatial';
```

STATUS muss VALID sein. Ist Spatial nicht installiert, kann es über den Oracle Installer problemlos hinzugefügt werden.

Oracle MapViewer

Oracle MapViewer stellt eine Anwendung um Oracle Application Server dar. Oracle bietet ein Standalone Kit, welches seht gut zur Entwicklung und für Tests genutzt werden kann. Die Installation ist denkbar einfach. Das ZIP File entpacken, die Pfade innerhalb der start.bat anpassen und die Datei dann ausführen.

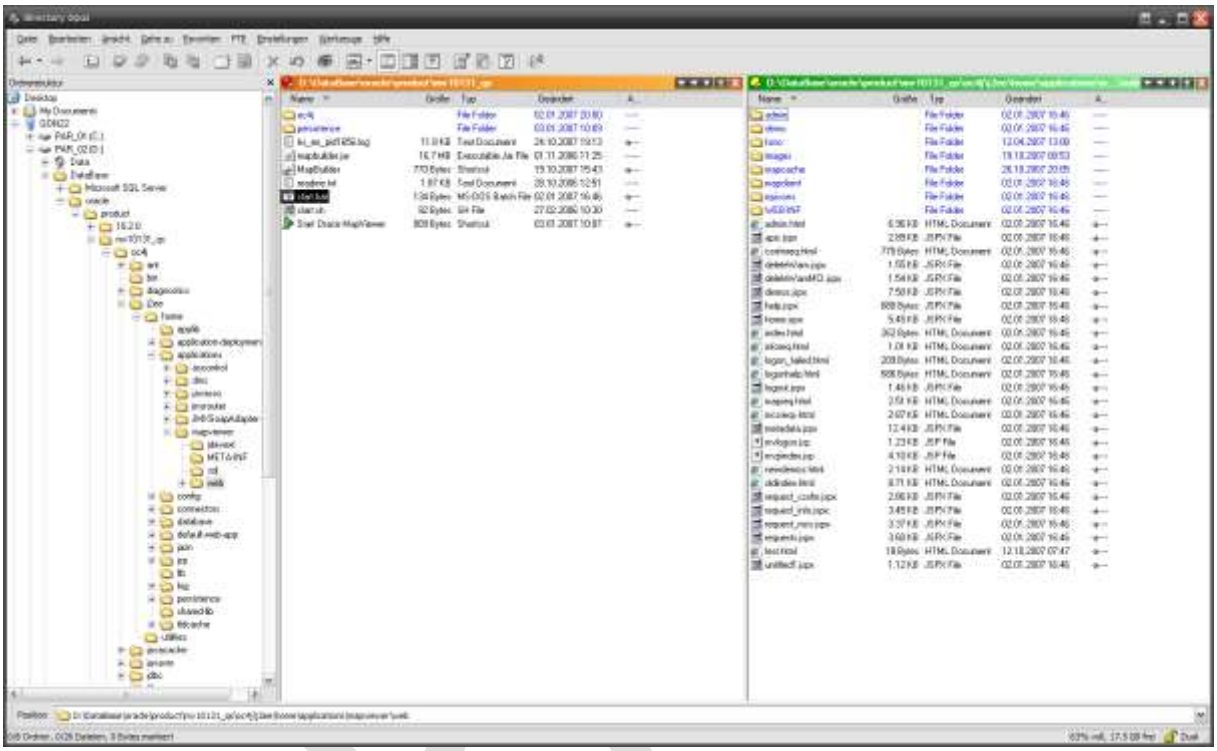

Abbildung 3: MapViewer Standalone Kit

Oracle MapViewer wird mit einer Instanz des Application Servers gestartet und ist unter Port http://localhost:8888/mapviewer verfügbar.

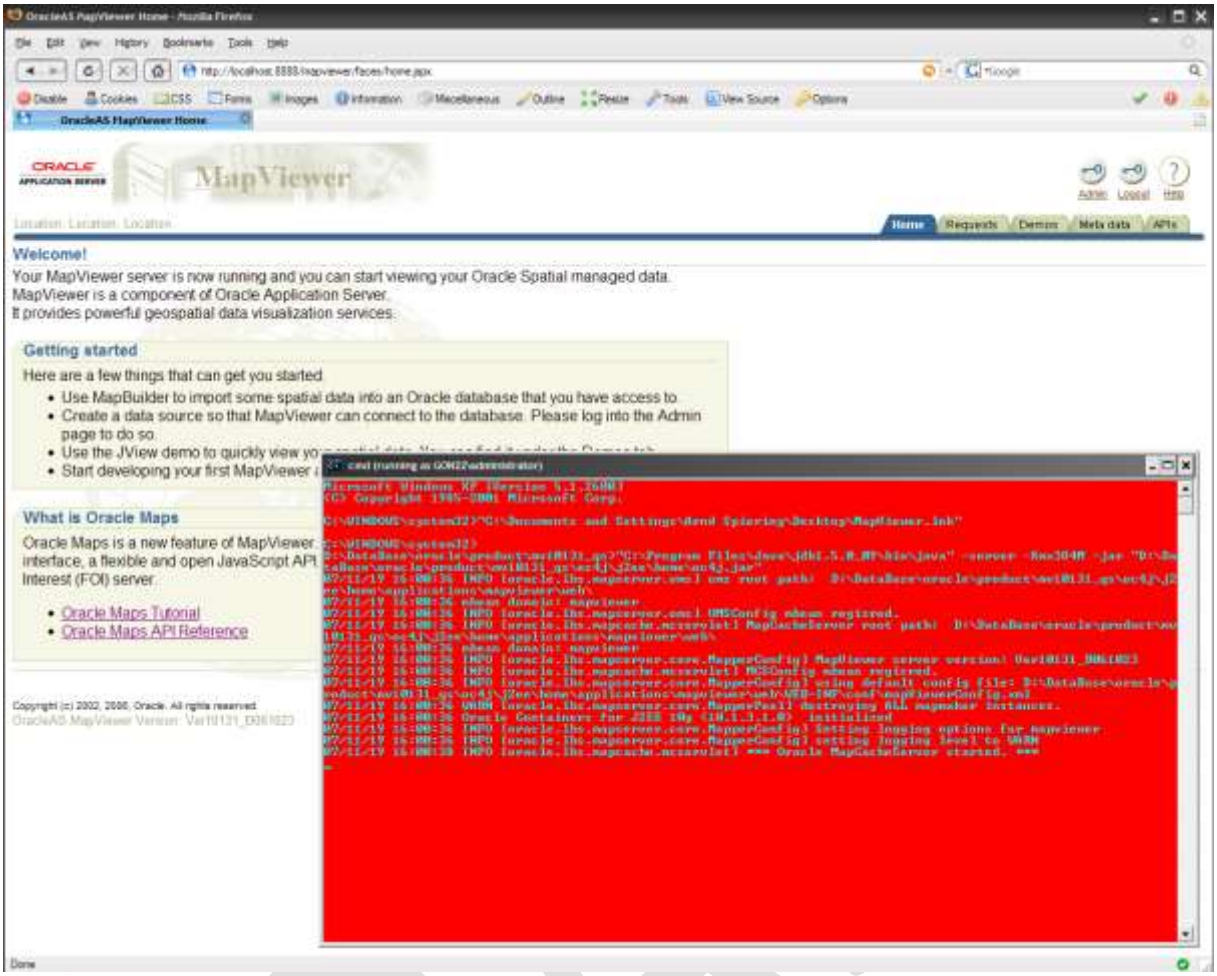

Abbildung 4: MapViewer Startseite im Hintergrund, im Vordergrund die DOS Konsole in der der OAS und MapViewer gestartet sind

Es sollte darauf geachtet werden, dass jetzt neben dem IIS ein zweiter WebServer (Apache) über Port 8888 aktiv ist. Auch dieser ist je nach Konfiguration von aussen erreichbar.

Oracle Maps

Oracle Maps wird direkt mit dem Oracle MapViewer mitgeliefert. In Abbildung 4 sind im linken Bereich weitere Informationen, die Dokumentation der API und Beispiele von Oracle Maps zu finden. Es muss nur die Konfiguration der MVDEMO Daten durchgeführt werden. Wichtig ist, dass die Konfiguration durchgeführt wird, wie es im Oracle Maps Bereich beschrieben ist. Die API Beispiele sind für einen ersten Einstieg sehr gut, weisen aber in einigen Bereich auch kleinere Fehler auf.

Oracle Proxy Server

An dieser Stelle werden jetzt 2 parallele Webserver betrieben, die voneinander nichts wissen. Führend ist an dieser Stelle ASP.NET, also der IIS Webserver. Oracle bietet einen IIS Proxy, der Request über ein Regelwerk an andere Webserver weiterleiten kann. Die Kommunikation findet aber immer nur zwischen dem Client und dem IIS statt, nie direkt mit dem anderen Weserver hinter dem Proxy.

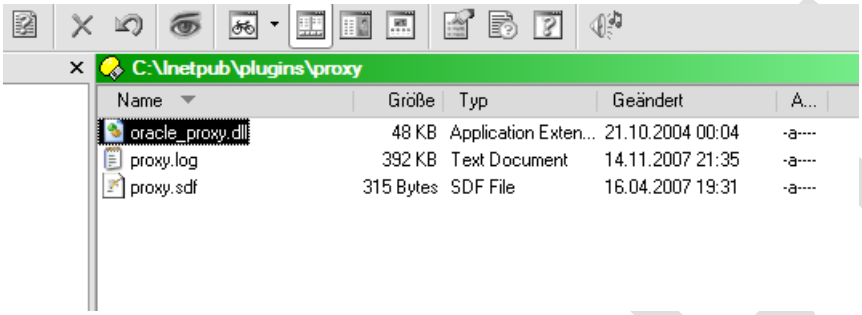

Abbildung 5: Oracle Proxy DLL

Die Konfiguration ist an sich gut dokumentiert. Die DLL wird ein einem Ordner abgelegt, ein neuer Registry Schlüssel wird manuell erzeugt und eine Regelwerk aufgesetzt. Danach wird der Proxy als ISAPI Erweiterung im IIS hinzugefügt. Hier ist zu beachten, dass beim IIS 6 diese Konfiguration direkt auf dem Knoten **WebSites** durchgeführt werden muss!

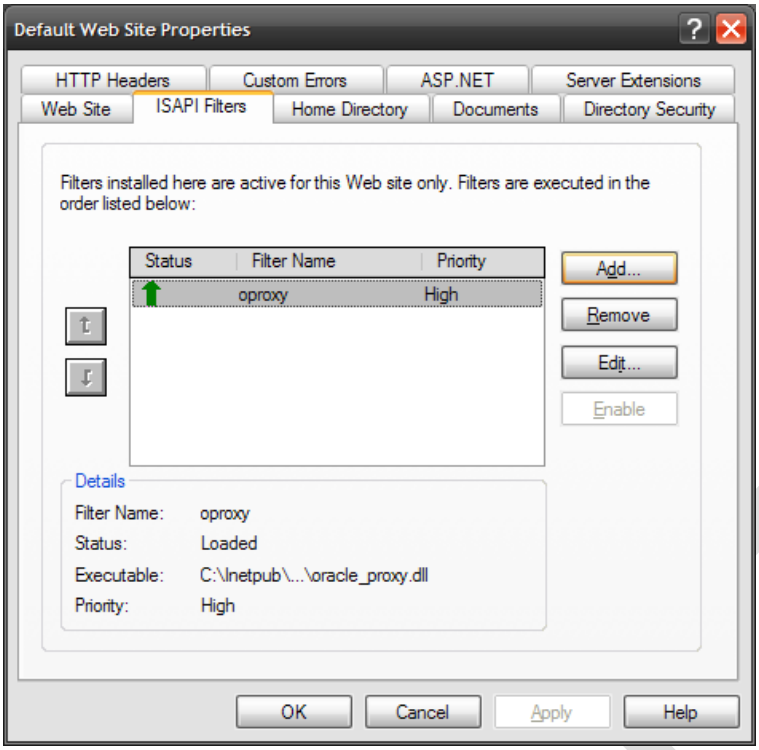

Abbildung 6: Oracle Proxy als ISAPI Erweiterung im IIS 5

Die Steuerdatei enthält das Regelwerk für die eigentlichen Weiterleitungen. Die Referenz zur Steuerdatei wird über die neuen Einträge in der Registry unter dem Schlüssel IIS Proxy Adapter erstellt.

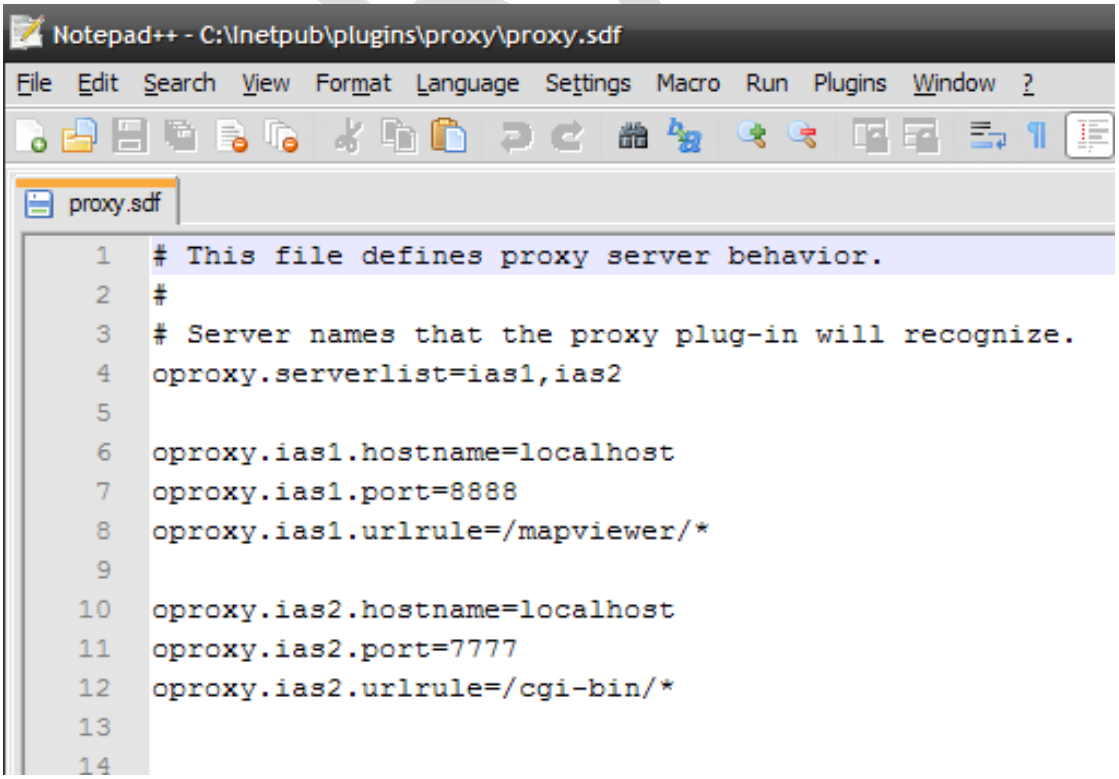

Abbildung 7: Konfigurationsdatei des Oracle Proxy mit 2 definierten Regeln, relevant ist nur die ias1 Konfiguration

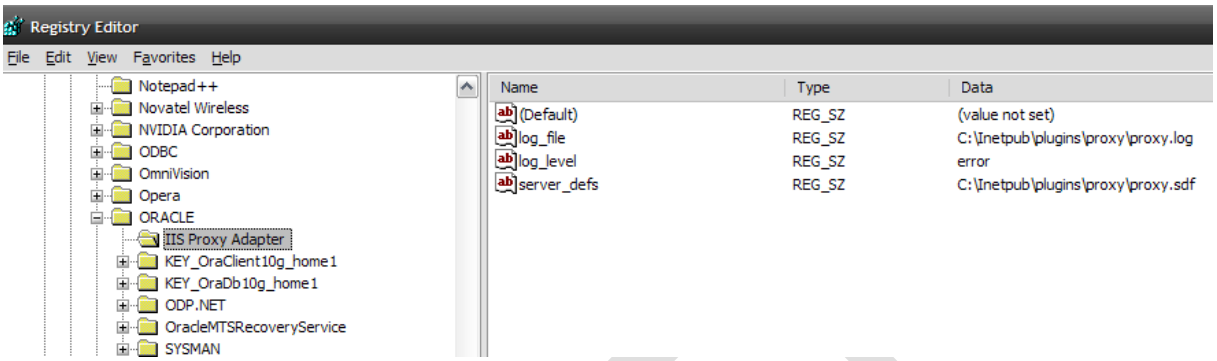

Abbildung 8: Neue Registry Einträge, die manuell erzeugt werden müssen

Nachdem alle Konfigurationen durchgeführt wurden, lässt sich die Funktionalität mit einer einfachen HTML Seite testen.

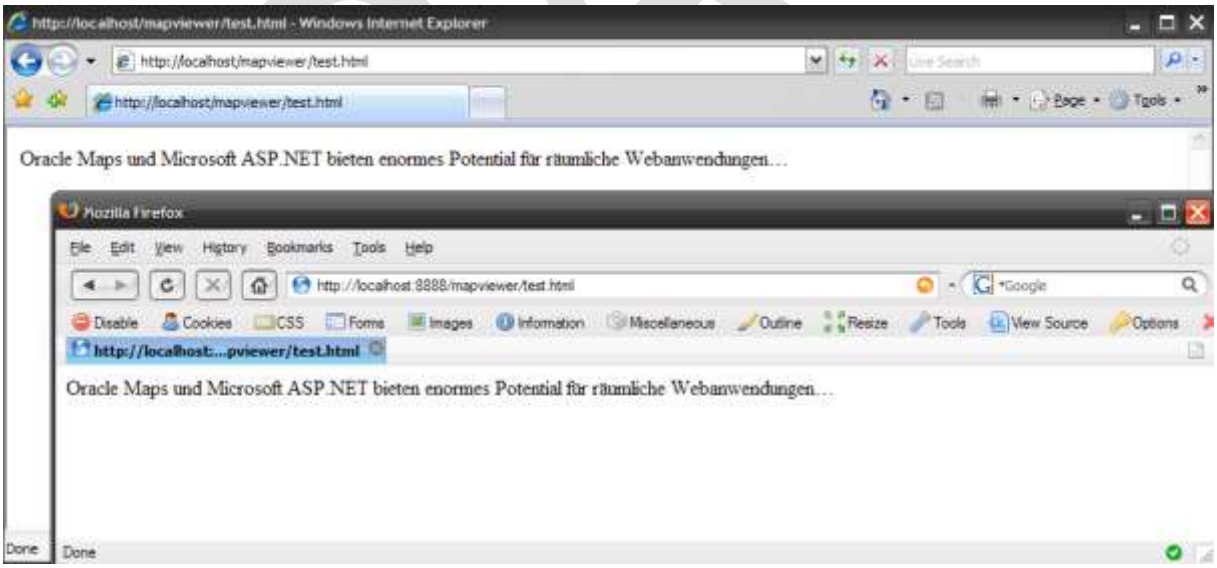

Abbildung 9: Zugriff mit und ohne Proxy, IE zeigt Zugriff über den IIS mit Proxy, Firefox zeigt Zugriff auf die Datei über den Apache Server

Ist es möglich, die Seite über Port 8888 und über Port 80 zu erreichen, ist die Konfiguration richtig ausgeführt worden.

Visual Studio Projekt mit Oracle Maps Integration

Bei der hier eingesetzten Visual Studio Version handelt es sich um VS2005 Professional. Das Projekt und die Entwicklung kann aber auch mit der kostenlosen Visual Web Developer Version realisiert werden.

Zunächst wird ein neues ASP.NET C# Projekt angelegt. Als Datenbasis dient die MVDEMO Konfiguration.

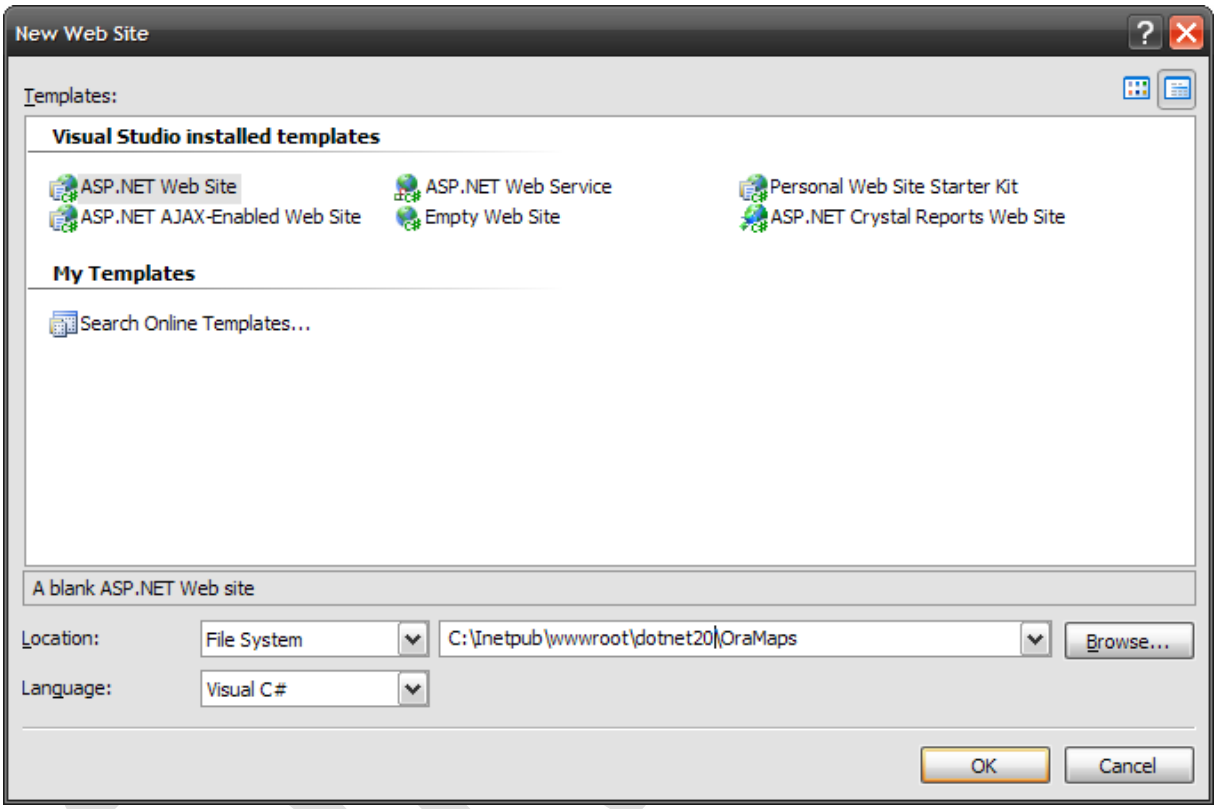

Abbildung 10: Neues Visual Studio Projekt

Das Projekt beinhaltet die Seite default.aspx in die Oracle Maps Funktionen implementiert werden. Die Datei erhält einen Javascript Referenz (script.js) für die benötigte Client Funktionalität und zum Einbinden der OracleMaps API. Wichtig ist, dass in den Eigenschaften des Projektes unter StartOptions UseCustomServer genutzt wird, damit überhaupt auf denn IIS zugegriffen wird. Weiter muss diese Datei eine Referenz zur oraclemaps.js Datei enthalten. Diese Datei bietet das eigentliche Ajax Interface.

Das OnLoad Event der Seite default.aspx ruft die Funktion showMap() in der Datei script.js auf.

```
var mapview;

// Initialisierungsfunktion
function showMap()
{
    var baseURL      = "http://localhost/mapviewer";
    var mapCenterLon = -122.45;
    var mapCenterLat =  37.6706;
    var mapZoom      =  3;
    var mpoint =
    MVSdoGeometry.createPoint(mapCenterLon,mapCenterLat,8307);
    mapview = new MVMapView(document.getElementById("map"), baseURL);
    mapview.addBaseMapLayer(new MVBaseMap("mvdemo.demo_map"));
    mapview.setCenter(mpoint);
    mapview.setZoomLevel(mapZoom);
    mapview.addNavigationPanel("EAST");
    mapview.display();
}
```

Die Funktion showMaps() nutzt Oracle Maps Funktionen, lädt die Inhalte der Karte nach und stellt diese in einem DIV Container dar.

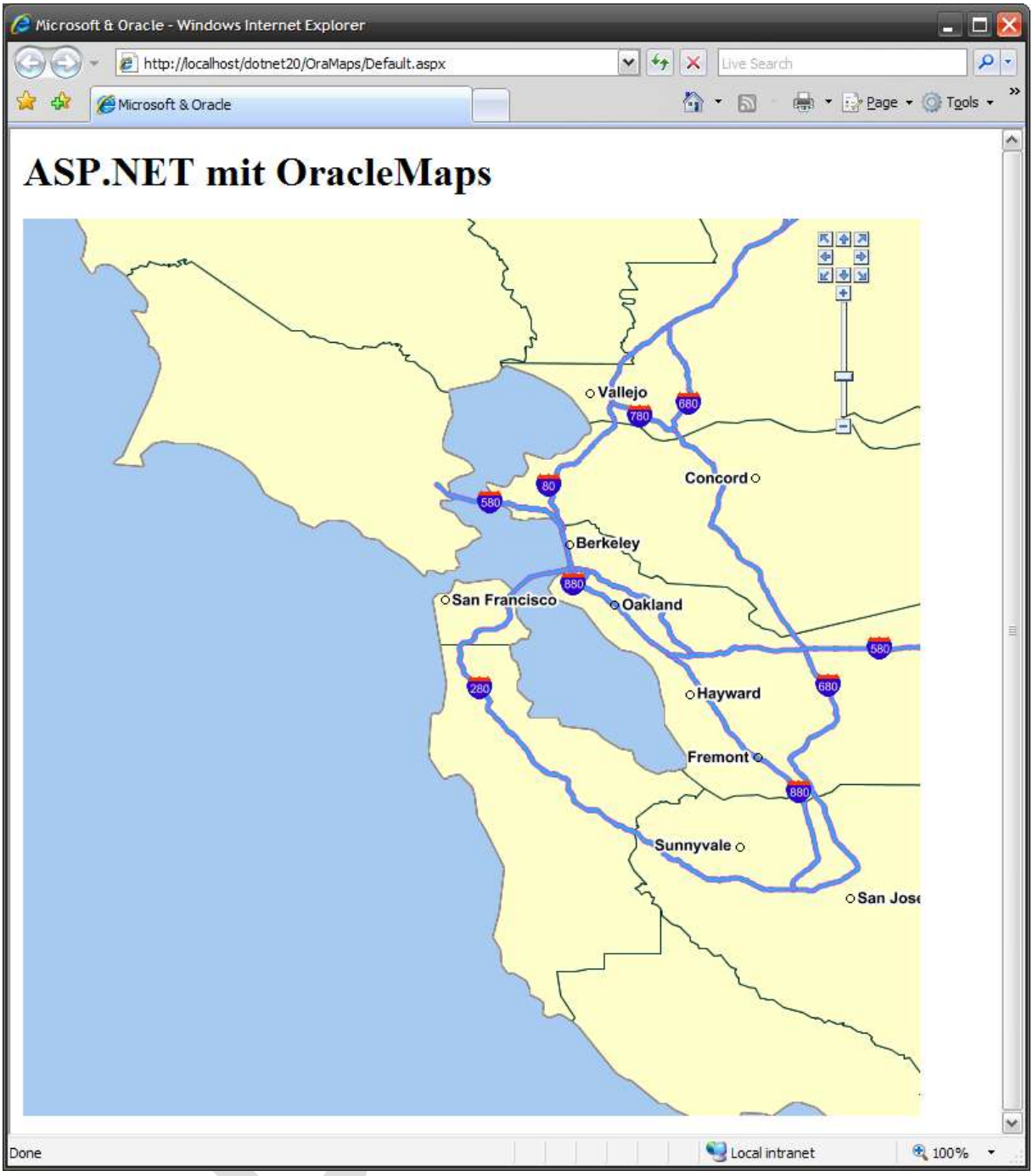

Abbildung 11: Oracle Maps Karte und GUI in einer ASP.NET Seite

Dynamische räumliche Abfragen aus der WebGUI

Zur Konfiguration der Metadaten von Oracle Maps kann (und sollte) die Anwendung MapBuilder genutzt werden. Mit dieser Anwendung werden Styles, Themes und BaseMaps definiert. Styles sind dabei graphische Ausprägungen, Themes Inhalte aus der Datenbank denen ein Style zugewiesen wird, und Base Map Kompositionen von Themes. Diese Konfigurationsanwendung bietet eine Möglichkeit, Oracle Maps mit der gesamten Palette von räumlichen Abfragen aus Oracle Spatial in Verbindung zu bringen.

Eine Kartendarstellung von Oracle Maps basiert auf einer oder mehreren BaseMaps, Themebased FOI Layern, User defined FOI Layer und weiteren Bestandteilen. Wir erzeugen an dieser Stelle UserDefined FOI Layer mit räumlichen Abfragen.

Das Beispiel:

Unsere Abfrage soll uns die nächsten CUSTOMERS zu einer INTERSTATE liefern. Dazu wird eine neues Theme NN_CUSTOMER_INTERSTATES erzeugt. Der Benutzer kann in der GUI die Nummer eines Highways angeben und die Anzahl der nächsten Kunden definieren. Es werden also 2 Parameter dynamisch über Bind Variables einer räumlichen Abfrage übergeben. Damit ist eine sehr gute Skalierbarkeit der Anwendung gegeben! Die eigentliche räumliche Abfrage wird als Query (unter Styling Rules) eines Geometry Themes definiert:

```
(mdsys.sdo_nn(LOCATION, (SELECT GEOM FROM MVDEMO.INTERSTATES WHERE HIGHWAY
= :1), 'sdo_num_res=' ||:2 ||' unit=m',1) = 'TRUE')
```

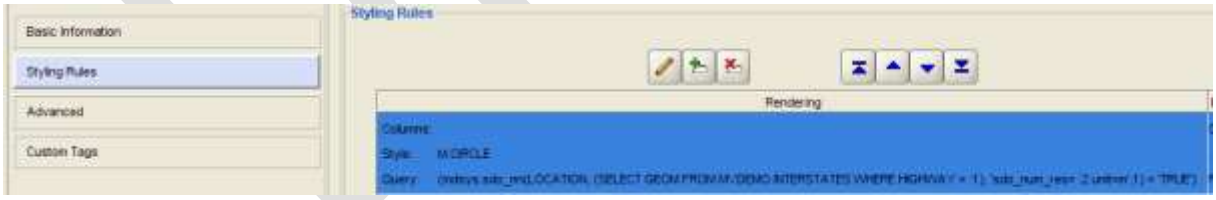

Abbildung 12: Definition der Query im Theme NN_CUSTOMER_INTERSTATE im Oracle MapBuilder

Die Abfrage kann durch Angabe der beiden Parameter auch direkt als Preview in Oracle Map Builder visualisiert werden. Konfigurationen sollten generell im Oracle MapBuilder überprüft werden, das verhindert eine komplizierte Fehlersuche innerhalb der Webanwendung. Auf der anderen Seite ist die Fehlersuche zu empfehlen, das bringt eine gewisse Übung….;-)

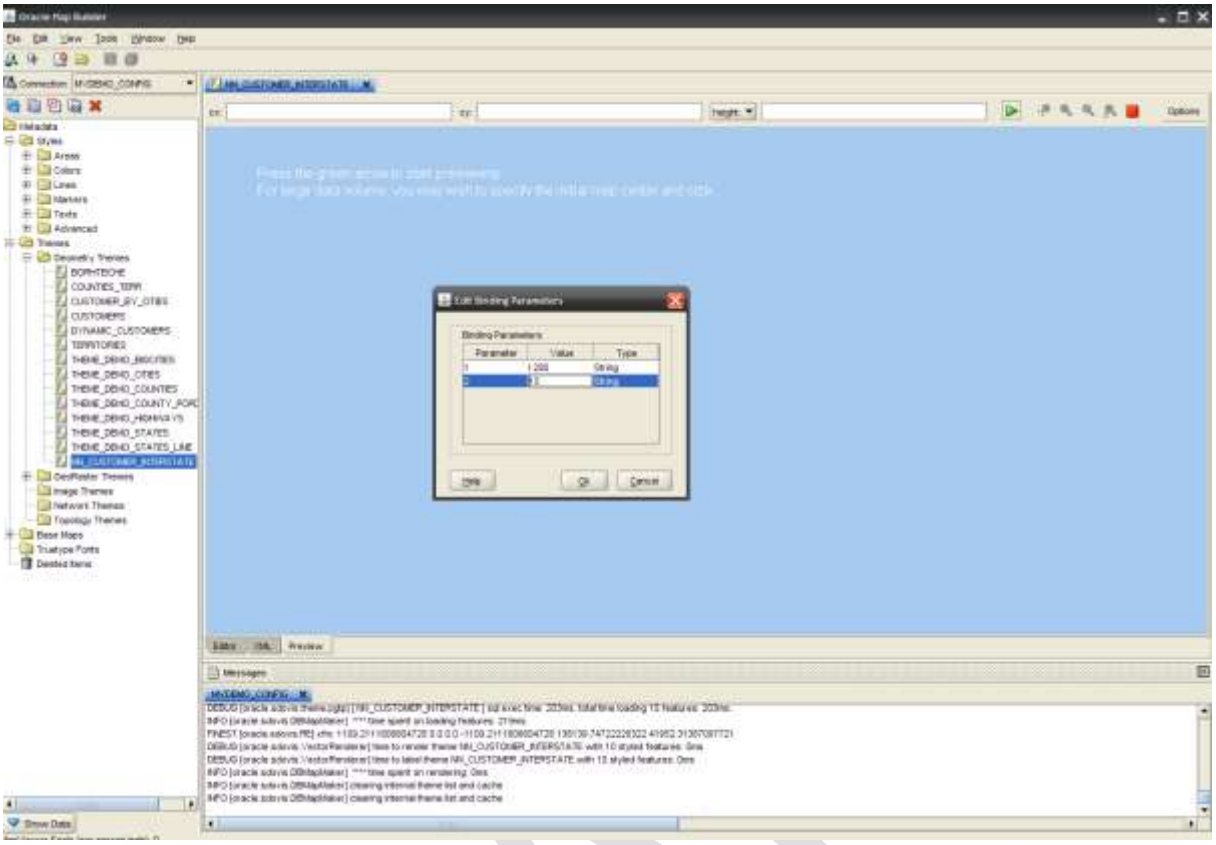

Abbildung 13: Start des Preview im Oracle MapBuilder, hier mit Angabe der Bind Variables

Die HTML GUI wird um drei Elemente erweitert. Zwei Textboxen werden für die Parameter und ein HTML Button zur Ausführung der Abfrage benötigt.
Das Click Event des HTML Buttons führt eine Java Script Funktion aus.

```
// Anzeigen eines Abfrageergebnisses
function dynQuery()
{
  try
  {
      var P = document.getElementById("TxtInterstate").value;
      var P2 = document.getElementById("TxtNumOfCustomers").value;

      var themebasedfoiCall04 = new MVThemeBasedFOI('ABFRAGE','NN_CUSTOMER_INTERSTATE');
      themebasedfoiCall04.setQueryParameters(P,P2);

      if (typeof mapview.getThemeBasedFOI('ABFRAGE') != 'undefined')
      {
          mapview.removeThemeBasedFOI(mapview.getThemeBasedFOI('ABFRAGE'));
      }

      themebasedfoiCall04.isVisible = true;
      themebasedfoiCall04.setBoundingTheme(true);
      mapview.addThemeBasedFOI(themebasedfoiCall04);
  }
  catch(e)
  {
      alert(e.description);
  }
}
```

Abbildung 14: JavaScript Funktion

Diese Funktion erzeugt einen FOI Layer basierend auf dem im MapViewer definierten Theme NN_CUSTOMERR_INTERSTATE. Die Abfrage wird zur Laufzeit ausgeführt und auf dem Client dargestellt.

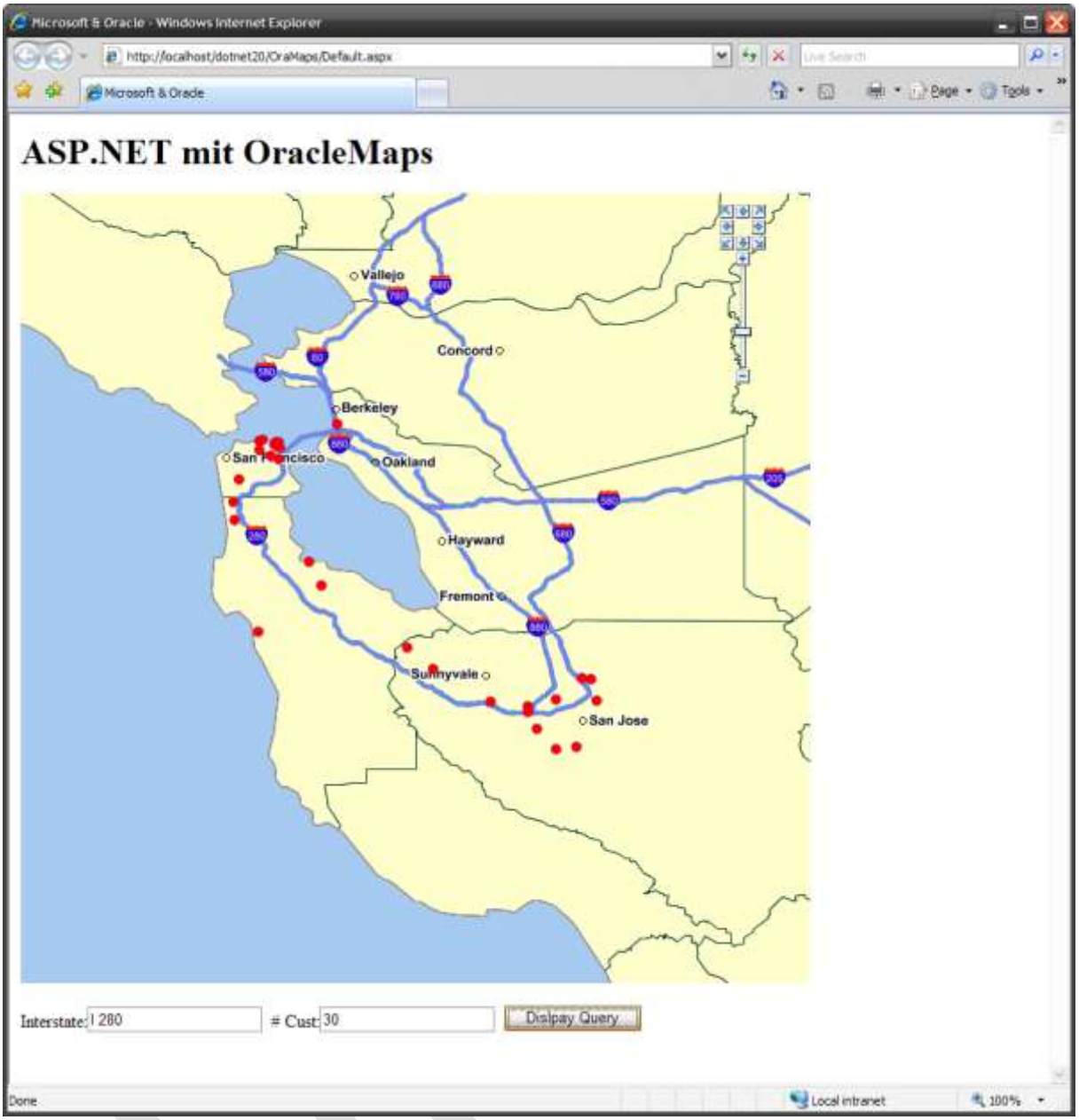

Abbildung 15: Abfrageergebnis als Kartendarstellung

Implementierung einer eigenen Ajax Funktionalität mit Oracle Maps Basisfunktionen

Als Beispiel einer eigenen Ajax Implementierung wird eine Koordinatenliste einer externen Seite geladen und die Karte in Intervallen auf die empfangenen Punkte zentriert.

Eine Java Script Funktion erzeugt ein XMLHTTP Objekt.

```
// XMLHTTP Objekt erzeugen
function GetXmlHttpObject()
{
  var xmlHttp=null;
  try
  {
      // Firefox, Opera 8.0+, Safari
      xmlHttp=new XMLHttpRequest();
  }
  catch (e)
  {
  // Internet Explorer
  try
  {
      xmlHttp=new ActiveXObject("Msxml2.XMLHTTP");
  }
  catch (e)
  {
      xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");
  }
  }
  return xmlHttp;
}
```

Abbildung 16: Erzeugen eines XMLHTTP Objektes

Der funktionale Ablauf wird in Bild 17 erläutert:

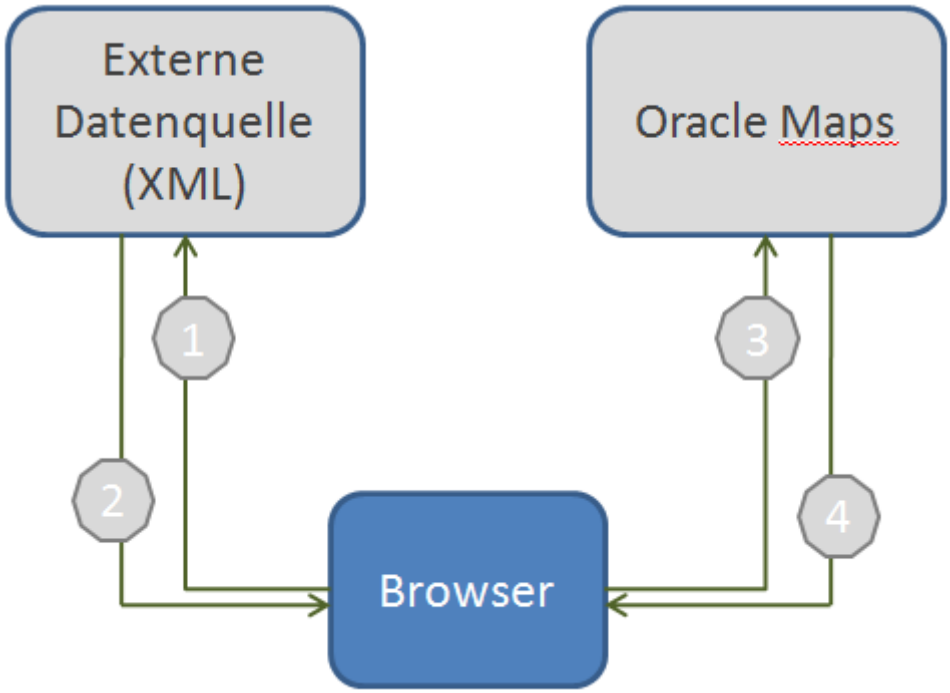

Abbildung 17: Darstellung des funktionalen Ablaufes

1. Das XMLHTTP Objekt führt einen GET Aufruf auf eine Datenquelle (hier eine ASPX Seite, die XML als Content Type liefert) durch
2. Das XMLHTTP Objekt empfängt den XML Datenstrom
3. Über die Oracle Maps API wird aus dem XML Datenstrom eine Point Geometry erzeugt
4. Die Karte wird auf den in Schritt 3 erzeugten Punkt zentriert

Zwischen Schritt 1 und Schritt 2 überwacht die Java Script Funktion stateChanged() den Status des XMLHTTP Objektes. Sind die XML Daten vom Server empfangen worden, wird die eigentliche Funktionalität zur Steuerung der Karte gestartet.
Es wird die JavaScript Funktion centerMap() aufgerufen, die ein Objekt vom Typ MVSdoGeometry erzeugt und den neuen Kartenmittelpunkt auf diesen Punkt festlegt.
Diese Funktion wird aus der Funktion stateChanged() zeitverzögert aufgerufen.

Fazit

Viele Unternehmen schätzen die Produktivität der .NET Entwicklungsumgebungen und die Stärken der Oracle Datenbank.

Es ist klar, dass die hier gezeigte Funktion nicht den Einsatz von Microsoft .NET fordert. Das Beispiel kann auch komplett Java basiert realisiert werden. Der Ansatz soll verdeutlichen, dass eine Erweiterung bestehender ASP.NET Anwendung mit Oracle Maps Komponenten ohne Probleme möglich ist.

Sie sollten diesen Ansatz während Konzeptphasen berücksichtigen.

Referenzen

Oracle Spatial, Locator and LocationBasedServices
http://www.oracle.com/technology/products/spatial/index.html

Oracle on .NET
http://www.oracle.com/technology/tech/dotnet/index.html

Microsoft ASP.NET
http://asp.net/

Oracle MapViewer Standalone Kit
http://www.oracle.com

Sourcen der Demo
http://www.gon.de/download/oramaps.zip

g.on experience gmbh
http://www.gon.de

Kontakt:
MAS GIS Arnd Spiering
arnd.spiering@gon.de
http://www.gon.de