

geo7

esriuserforum.ch 2009 PromoPost-Tool

Pierre Honsberger, geo7 AG
12.03.2009

Inhalte

- Produktinfo
- Karten
- GIS-Funktionalität
- Architektur
- Implementation
- Implementationsprobleme
- Implementationsverbesserungen/-optimierungen
- Migration auf AGS 9.3
- Performance
- Projektumfeld
- Zusammenfassung

Produktinfo

- Einstiegsseite: <http://www.promopost.ch>
- **Originalzitat Post:**
Unadressierte Direktwerbung (PromoPost) – gezielt neue Märkte erschliessen
- **Erfassung von Mailingaufträgen und Berechnung der anfallenden Kosten**
- **Aktuelle Version: ohne GIS-Anbindung**

Promopost: neue Version

- **Neu ab Frühling 2009: zusätzlich mit GIS-Modul**
- **2 Module**
 - Business: Smartdynamic (80%)
 - Gis: geo7 AG (20%)
- **vorläufige Einstiegsseite:**
<https://promopostint.post.ch/ppt/>
- **Einbettung des GIS-Modul in der Business-Applikation als IFRAME**
- **ASP.NET (C#) Applikation auf Basis von AGS 9.2**

Karten

- Darstellung von 8 Selektionsebenen (Gebiete)
 - Land, Kanton, Amtsbezirk, Gemeinde
 - WEMF-Region, WEMF-Gebiet, WEMF-Agglo
 - Postleitzahlen
- ➔ 8 Map Services im AGS (auf der Basis von .mxd Dokumenten) publiziert
- Im Hintergrund werden immer die Postleitzahlen ‚mitberechnet‘
- **Massgebend für die Kostenberechnung ist immer eine Auswahl von Postleitzahlen**

GIS Funktionalität

- Selektion von Gebieten über:
 - Punkt
 - Polyline
 - Polygon

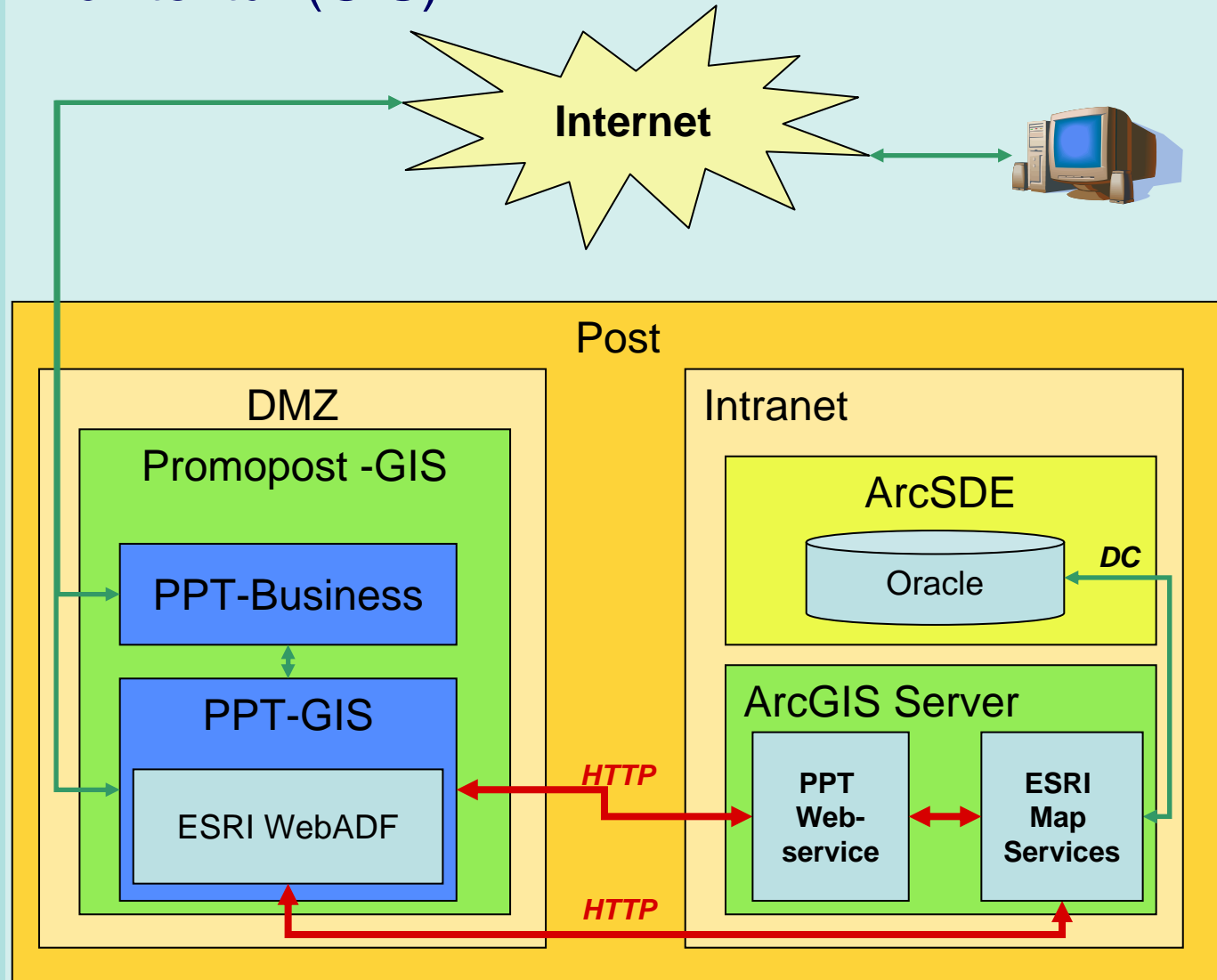
➔ Spatial Queries
- Selektion mit / ohne Buffer
- ➔ Spatial Queries
- Übliche GIS- Funktionalität
 - Zoom
 - Pan
 - Full Extent
- Weitere Selektionskriterien
 - Sprache (D, F, I)
 - Tarifkategorien (A, B, C, D)

➔ Attribute Queries

Einfache GIS Selektionsbeispiele

- Alle Gemeinden im Aaretal Bern bis Thun
 - Umkreis 0 km
 - Polyline Bern bis Thun
- Alle Gemeinden im Umkreis von 5 km im Aaretal Bern bis Thun
 - Umkreis 5 km
 - Polyline Bern bis Thun
- Alle Gemeinden (ohne Städte) im Umkreis von 5 km im Aaretal Bern bis Thun
 - Umkreis 10 km
 - Polyline Bern bis Thun
 - Tarifkategorie B,C,D (ohne A)

Architektur (GIS)



Architektur (GIS): Rahmenbedingungen

- Webapplikation und ArcGIS Server sind getrennt.
- In der Webapplikation ist das ESRI Web ADF Framework installiert, jedoch nicht ArcObjects (nur in AGS).

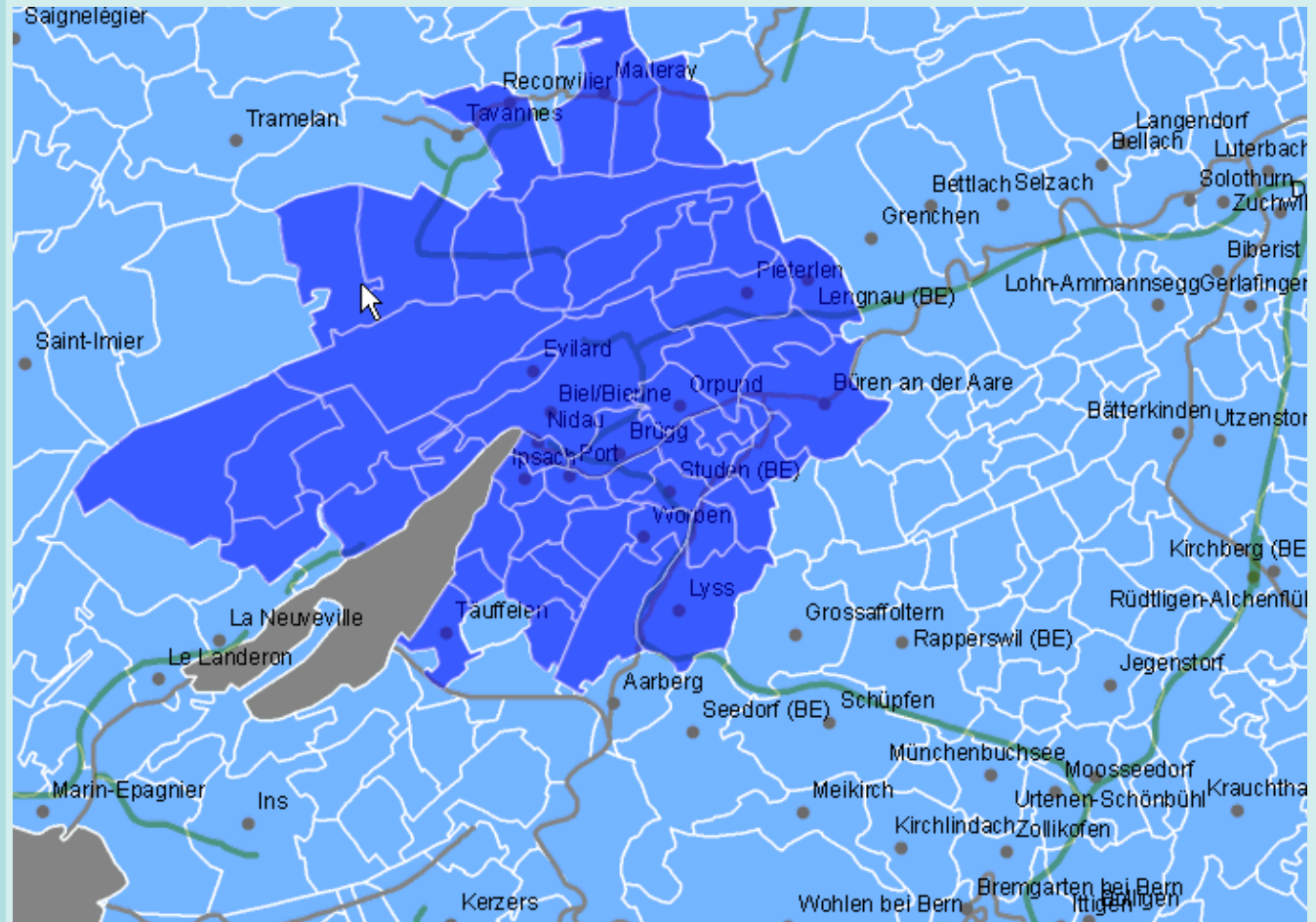
Benötigte GIS-Funktionalität

- besteht aus 3 Grundoperationen
- Kartendarstellung und Navigation (Pan / Zoom, / Full Extent) wird **automatisch über das WebADF** realisiert. Dazu wird (implizit) die Methode ExportMapImage des Map Services verwendet.
- Auswahl (Query) von Selektionsebenen (Polygone) aufgrund der Userinteraktionen ist mit den Methoden des Map Services QueryFeatureData möglich (**aber... siehe weiter unten!**).
- Einfärben (**Highlight**) der selektierten Polygone wird mit Web ADF Funktionalität realisiert (Graphics Layers).

Eine typische Aufgabe

- User möchte eine Gemeinde auswählen und im Umkreis (Buffer) von 5km alle weiteren Gemeinden selektieren. Als Resultat sind diese Gemeinden einzufärben.

Eine typische Aufgabe



Workflow

- Gemeinde auswählen durch Klicken (Web)
- X/Y Koordinate des Clicks bestimmen und damit Gemeinde selektieren (mit Spatialfilter und Intersect)
(Web → AGS → Web)
- Buffer um Gemeinde generieren (?)
- Mit Buffer nochmals die betroffenen Gemeinden bestimmen (Web->AGS->Web)
- Die Gemeinden einfärben (Web)
- Mit den selektierten Gemeinden die betroffenen Postleitzahlen bestimmen (mit Spatialfilter und Intersect)
(Web → AGS → Web)
- bedingt einige Interaktionen schon für eine einfache Fragestellung ➡ Auswirkungen auf Performance?

Implementation: 1. Lösungsansatz

- Ausschliesslich QueryFeatureData benutzen
- Jeder ESRI Map Service bietet diese Funktionalität über seine Schnittstellen an
- QueryFeatureData benötigt einen Queryfilter (Spatial oder Attribut) und gibt ein DataSet zurück
- Das Resultat enthält (optional) die Geometrien
- Diese Geometrien werden zum Einfärben der selektierten Gemeinden benötigt
- **Wichtige Konsequenz: ➡ Zwischen Web und AGS müssen die Geometrien ausgetauscht werden**

Implementation: Probleme

- Dieser 'QueryFeatureData' Lösungsansatz hat 2 grundsätzliche Probleme
- Bufferfunktionalität ist noch nicht gelöst.
- Performanceeinbussen durch wiederholte Übertragung von (umfangreichen) Geometrien (und erst noch in SOAP-Envelopes).
- Veranschaulichung: Auswahl von 600 Gemeinden (im Hintergrund mindestens 600 Postleitzahlen) ➔ **mindestens 1200 Geometrien müssen ausgelesen und übertragen werden.**

Implementation: Verbesserungen

- Verbesserungen:
- Fehlende Bufferfunktionalität mit **eigenem Webservice auf dem AGS** lösen (dadurch zusätzliche Optimierungen). Dieser Webservice nutzt selbst ArcObject Funktionalität des AGS.
- Performanceproblematik entschärfen durch **Übertragen minimaler Information** zwischen der Webapplikation und dem ArcGIS Server
- (Mögliche weitere Optimierung durch Attribut- anstatt SpatialQueries ist nicht möglich, da in den Featurelayer von ArcSDE die Attribute nicht enthalten sind.)

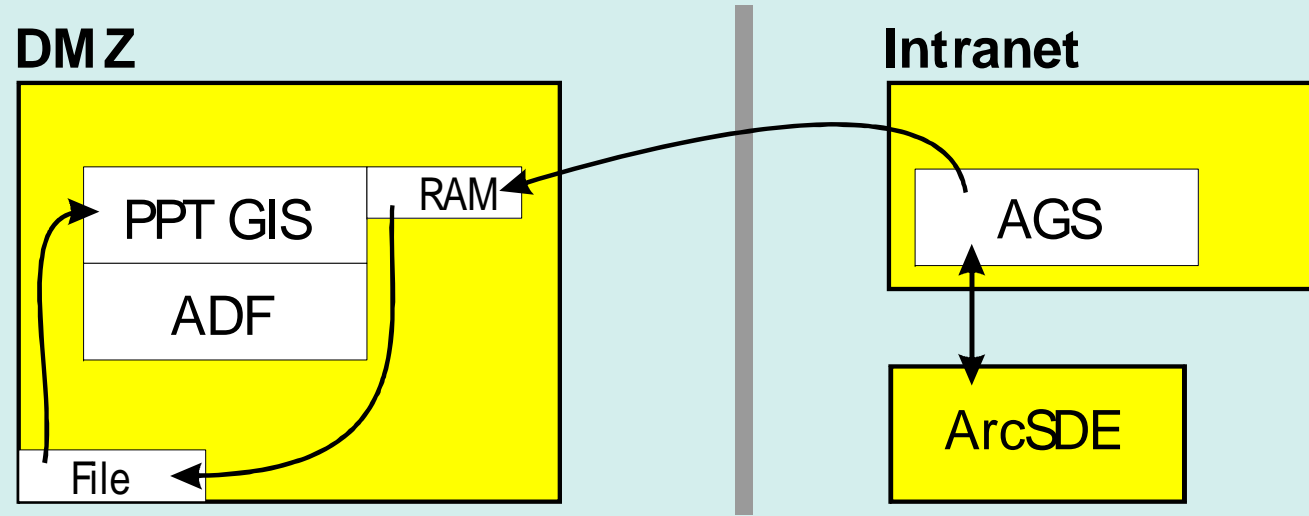
Implementation: Verbesserungen

- Eigener Webservice
 - Anstatt direkt nur die von ESRI angebotenen Map/Webservices zu benutzen, werden **eigene optimierte Methoden** angewendet. Diese führen auf dem AGS die vielen Operationen hintereinander aus und geben erst zuletzt das gewünschte Resultat zurück.
- Übertragen minimaler Informationen
 - Anstatt bei jeder Anfrage immer wieder die Geometrien zu übertragen, werden **nur ihre ID's** übermittelt.
 - Dadurch wird aber **neu erforderlich**, dass die Webapplikation lokal die Geometrien im Zugriff hat. Dazu wird beim Aufstarten der Applikation **einmalig ein Cache mit allen Geometrien generiert**. Mit den ID's können diese danach sehr effizient aus dem lokalen Cache ermittelt werden und auf der Karte eingefärbt werden.

Weitere Optimierungen

- Simplify auf Geometrien (Einsparung bis 75%) vornehmen.
- Für die Kartendienste ESRI-Mapcaches aufbauen.
- Lokaler Geometrie-Cache von serialisierten Files aufbauen, anstatt diese zu Beginn vom AGS zu holen (siehe Abbildung).

Lokaler Geometrie-Cache



Migration auf AGS 9.3

- Zurzeit wird die Applikation auf AGS 9.3 migriert. Diese Migration erfordert gewisse zusätzliche ursprünglich nicht geplante Schritte. Hauptgründe sind:
 - Umstellung auf neue Assemblies des Frameworks in VS 2008.
 - Codeänderungen wegen neuem WebADF Framework.
 - Codeänderungen wegen neuen Erweiterungen im clientseitigen AJAX/JSON Framework.
- Neue Funktionalität ➡ neue Möglichkeiten
 - Geometry Service bietet neue Lösungsansätze für die Bufferfunktionalität.
- Behobene Fehler
 - Problem von falscher Darstellung von Enklaven bei Selektionen (z.B. St.Gallen und Appenzell) ist nun gelöst.

Performance

- Von Anfang an war die „Performance“ ein vom Kunden immer wieder erwähnter Punkt.
- Unter „Performance“ verstand man aber im Wesentlichen die nötige Zeit für die Kartendarstellung, nicht die für die eigentlichen Funktionen.
- **Performance ist inzwischen sehr gut.**

Projektumfeld

- Das Projektumfeld bei ArcGIS Server Lösungen ist wesentlich komplexer und bedarf viel mehr Einsatz bei der Koordination
 - Grosse und komplexe Rechenzentren
 - erhöhte Sicherheitsansprüche
 - Zuverlässigkeit
 - Diverse Vorgaben an die Applikationen
- Viele IT-Spezialisten haben kein GIS-Know How und auch kein Verständnis, wenn wieder mal ein Dienst neu gestartet werden muss...
- Schwierigkeiten bei der Abgrenzungen bei auftretenden Problemen.

Zusammenfassung

- Installation von Webapplikation und ArcGIS Server auf getrennten Plattformen führen dazu, dass die Performancefragen sehr schnell akut und gelöst werden mussten.
- Web ADF ist grundsätzlich brauchbar, muss aber für spezielle Problemlösungen durch eigene Module (Web Services) erweitert werden
- Migration auf höhere ArcGIS Versionen sind selten problemlos zu realisieren

Fragen?

**Vielen Dank für Ihre
Aufmerksamkeit!**