|  |
| --- |
|  |
| / Systemdesign |
|  |

|  |  |
| --- | --- |
| Klassifizierung \* | Nicht klassifiziert |
| Status \*\* | In Arbeit |
| Projektname | MISTRA-Fachapplikation Erhaltungsmanagement im Siedlungsgebiet |
| Projektabkürzung | MISTRA-EMSG |
| Projektleiter |  |
| Auftraggeber | Bundesamt für Strassen – ASTRA |
| Autoren |  |
| Initiale | TP, PV |
| Bearbeitende | <name>, <name> (Realisierung) |
| Prüfende | <name>, Gesamtprojektleitung MISTRA (GPL Stv.)  <name>, ASTRA-Architekt  <name>, Leitung Realisierung (PL Stv. LR)  <name>, Leitung Realisierung  <name>, Projektleiter (PL)  <name>, ASTRA-GIS-Architekt |
| Genehmigende | <name>, Projektleiter (PL) |
| Verteiler | LR, EL, GPL, <name> (IT-Architekturmanagement ASTRA), <name> (BIT) |
| Doc\_ID |  |
| Kurzbeschreibung | Dieses Dokument entspricht den Anforderungen an Systemarchitektur und Systemdesign laut HERMES für das MISTRA-Projekt EMSG. |

\* Nicht klassifiziert, Intern, Vertraulich

\*\* In Arbeit, In Prüfung, Abgeschlossen

Änderungskontrolle, Prüfung, Genehmigung

| Version | Datum | Beschreibung, Bemerkung |
| --- | --- | --- |
| 0.1 | 08.09.2011 | Entwurf |
| 0.75 | 30.09.2011 | Inhaltliche Ergänzungen |
| 0.77 | 30.09.2011 | Qualitätssicherung |
| 0.78 | 19.10.2011 | Neues Template (von <name>) |
| 0.79 | 20.10.2011 | Arbeitspakete für Ergänzungen |
| 0.151 | 18.11.2011 | Vollständige Überarbeitung |
| 0.152 | 18.11.2011 | Qualitätssicherung |
| 0.153 | 05.12.2011 | Konsolidierte Review seitens EL, GPL, ASTRA-Architekten und LR |
| 0.154 | 06.12.2011 | Formelle Überarbeitung |
| 0.158 | 12.12.2011 | Inhaltliche Überarbeitung laut Review |
| 0.170 | 15.12.2011 | Ergänzungen Achsenaktualisierung |
| 0.171 | 15.12.2011 | Qualitätssicherung |
| 0.173 | 22.12.2011 | Review des Abschnitts 4.3.2. hinsichtlich der bedingten Abnahme |
| 0.192 | 13.01.2012 | Überarbeitung der noch offenen Punkte |
| 0.193 | 13.01.2012 | Qualitätssicherung |
| 0.194 | 16.01.2012 | Logisches Netzwerk und Datenflüsse aktualisiert |
| 0.195 | 27.01.2012 | Review LR |
| 0.196 | 29.01.2012 | Integration Review ASTRA-Architekten und Bereinigung der Kommentare mit EL sowie Gesamtqualitätssicherung |
| 0.197 | 27.01.2012 | Abschnitt 7.1.1 (Achsenaktualisierung) ergänzt |
| 0.199 | 01.02.2012 | Überarbeitung nach Review LR |
| 0.200 | 06.02.2012 | Abschnitt 4.6.4.6 (Berechnungslogik) ergänzt |
| 0.202 | 28.02.2012 | Einarbeitung Feedback LR/EL |
| 0.205 | 01.03.2012 | Achsen-Import und Achsen-Update: Anpassung Text; Aktualisierung ERM |
| 0.208 | 26.04.2012 | Überarbeitung Achsenupdate, Beantwortung Kommentare |
| 0.209 | 01.05.2012 | Qualitätssicherung |
| 0.213 | 31.05.2012 | Ergänzungen und Korrekturen: Reporting GIS, Achsen-Import, Mandantenfähigkeit |
| 0.221 | 17.07.2012 | Aktualisierung ERM und diverse Ergänzungen |
| 0.222 | 24.07.2012 | Ergänzungen und Korrekturen: Reporting GIS, Achsen-Import, Mandantenfähigkeit |
| 0.228 | 01.08.2012 | Online-Hilfe, Benchmarking und andere Ergänzungen |
| 0.230 | 02.08.2012 | Ergänzungen GIS |
| 0.234 | 02.08.2012 | Integration Technische Dokumentation |
| 0.236 | 03.08.2012 | Ergänzungen GIS, EMSG-Mobile |
| 0.248 | 07.08.2012 | Ergänzungen Technische Dokumentation |
| 0.250 | 21.08.2012 | Ergänzungen Technische Dokumentation |
| 0.251 | 27.08.2012 | Erledigte Kommentare entfernt |
| 0.256 | 30.08.2012 | Ergänzungen Technische Dokumentation |
| 0.257 | 31.08.2012 | Qualitätssicherung |
| 0.258 | 03.09.2012 | Inhaltsverzeichnis repariert |
| 0.259 | 29.10.2012 | Zoomstufen angepasst |
| 0.261 | 13.12.2012 | Erweiterung für Änderungsantrag 04 |
| 0.262 | 21.12.2012 | Review-Kommentare eingearbeitet |
| 0.263 | 23.12.2012 | Review-Kommentare eingearbeitet |
| 0.264 | 02.01.2013 | Review-Kommentare eingearbeitet |
| 0.265 | 02.01.2013 | Qualitätssicherung |
| 0.266 | 06.02.2013 | Review LR |
| 0.268 | 15.02.2013 | Review EL |
| V1.3.0.0 | 18.02.2013 | Version für BIT-Deployment |
| 0.274 | 11.03.2013 | Überarbeitung laut Review LR/EL |
| 0.275 | 12.03.2013 | Überarbeitung laut Review LR/EL |
| 0.276 | 13.03.2013 | Überarbeitung laut Review LR/EL |
| 0.279 | 21.03.2013 | Überarbeitung laut Review LR/EL |
| 0.281 | 26.03.2013 | Überarbeitung laut Telko LR/EL |
| 0.282 | 02.04.2013 | Überarbeitung laut Review LR/EL |
| 0.283 | 04.04.2013 | Überarbeitung laut Review LR/EL |
| 0.284 | 09.04.2013 | Architektur-/Datenflussskizze aktualisiert |
| V1.3.0.1 | 15.04.2013 | Version für BIT-Lieferung |
| 0.290 | 17.04.2013 | Überarbeitung laut Review LR/EL |
| 0.291 | 18.04.2013 | Einsatz und Verteilung ergänzt |
| 0.292 | 19.04.2013 | Qualitätssicherung |
| 0.297 | 26.06.2013 | Überarbeitung aut Review LR/EL |
| 0.298 | 02.07.2013 | Version für Werkabnahmeprotokoll |
| 1.0 | 02.07.2013 | Freigabe |
| V1.4.0.0 | 05.06.2014 | Anpassung laut Änderungsantrag 05 |
| V1.5.0.0 | 02.12.2014 | Abschnitt 5.2.1.1 (Webservices Basissystem) korrigiert |
| V1.6.0.0 | 23.02.2016 | Versionsnummer aktualisiert |
| V1.7.0.0 | 03.04.2017 | Anpassung laut Änderungsantrag 08, Änderungsantrag 09 und Änderungsantrag 10 |

Inhaltsverzeichnis

Änderungskontrolle, Prüfung, Genehmigung 2

Inhaltsverzeichnis 5

Abbildungsverzeichnis 8

1 Allgemeines 10

1.1 Begriffe 10

1.2 Referenzierte Dokumente 12

1.3 Management Summary 12

2 Zweck des Dokuments 13

2.1 Aufgabenstellung 13

2.2 Architekturziele 13

2.3 Im EMSG-Master kann der Endanwender vordefinierte Reports (Microsoft Reporting Service) ausführen und das Ergebnis auf seinen Client herunterladenRahmenbedingungen der Architektur 13

3 Lösungsvorschläge für die Struktur des Systems 14

4 Struktur des Systems 15

4.1 Logische Sicht 15

4.1.1 Präsentationsschicht (Clients) 15

4.1.2 Geschäftslogikschicht 16

4.1.3 Datenschicht 16

4.2 Anwendungsfälle-Sicht 17

4.3 Prozess-Sicht 18

4.3.1 Jahresabschluss in EMSG-Master 18

4.3.2 Aktualisierung der Achsen im EMSG-Master 21

4.3.3 Benchmarking 31

4.4 Daten-Sicht 34

4.4.1 Aufbau von Tabellen im Datenmodell 34

4.4.2 Kardinalität 34

4.4.3 Namenskonvention 35

4.4.4 Modusübergreifende Tabellen 36

4.4.5 Summarischer Modus 41

4.4.6 Strassennamen Modus (tabellarisch) 42

4.4.7 GIS-Modus 44

4.4.8 Inspektionsrouten und Check-Out von Daten 55

4.4.9 Datenmodell EMSG-Mobile 56

4.4.10 Datentypen 58

4.5 Deployment-Sicht 58

4.5.1 Skalierbarkeit 58

4.5.2 Einsatz und Verteilung 58

4.5.3 Parametrisierung 59

4.5.4 Netzwerkzonen 59

4.6 Realisierungs-Sicht 61

4.6.1 Schichten des Systems EMSG-Master 61

4.6.2 Interaktion der verwendeten Komponenten in EMSG-Master 66

4.6.3 Schichten des Systems EMSG-Mobile 69

4.6.4 Interaktion der verwendeten Komponenten in EMSG-Mobile 70

4.6.5 Code-Sicht 72

4.6.6 Benutzeroberfläche EMSG-Master 72

4.6.7 Benutzeroberfläche EMSG-Mobile 78

4.7 Infrastruktur- und Hardware-Sicht 78

4.7.1 Verwendete Produkte und Methoden 78

4.8 Design-Strategien EMSG-Master 83

4.8.1 Locking 83

4.8.2 Validierung 83

4.8.3 Anwendungslog 83

4.8.4 Audit Log 84

4.8.5 Überwachung 84

4.8.6 Fehlerbehandlung 84

4.8.7 Mehrsprachigkeit 84

4.8.8 Hilfe 87

4.9 Design-Strategien EMSG-Mobile 88

4.9.1 Locking 88

4.9.2 Validierung 88

4.9.3 Logging 88

4.9.4 Überwachen 89

4.9.5 Fehlerbehandlung 89

4.9.6 Mehrsprachigkeit 89

4.9.7 Hilfe 89

5 Schnittstellen 90

5.1 Interne Schnittstellen 90

5.1.1 Karte (GeoServer) 90

5.1.2 Datenaustausch EMSG-Master mit EMSG-Mobile 92

5.2 Externe Schnittstellen 95

5.2.1 MISTRA Basissystem 96

5.2.2 WMS – swisstopo 96

5.2.3 WMTS – swisstopo 97

5.2.4 Mengengerüst GIS-Schnittstellen 97

6 Anforderungszuordnung 98

7 Realisierbarkeitsuntersuchungen 98

7.1 Prototypen 98

**7.1.1** **Achsenaktualisierung** 99

7.2 Referenzprodukte 100

7.2.1 Geoshop 100

7.2.2 JaNo 101

8 Sicherheit 102

8.1 Schutzmassnahmen 102

8.2 Benutzer- und Rechteverwaltung 102

8.2.1 Authentifizierung 102

8.2.2 Autorisierung und Datensicherheit 102

9 Wartung 107

9.1 Quellcode 108

9.1.1 Common 108

9.1.2 Master 108

9.1.3 Mobile 109

9.2 Wichtige Konfigurationsmöglichkeiten 110

9.3 Einrichten der Entwicklungsmaschine 110

9.3.1 Entwicklungsumgebung 110

9.3.2 IIS 110

9.3.3 Frameworks 110

9.3.4 Tools 111

9.4 Unit-/Integration Tests 111

9.5 Developer Client 111

9.5.1 Datenbankschema 112

9.5.2 Erzeugen von Stammdaten 113

9.5.3 Bearbeiten der Berechtigungen 113

9.5.4 Erzeugen von Testdaten 114

9.5.5 Deployment Package 114

9.5.6 SQLiteDb 115

9.5.7 Test Data Import 115

9.6 Physisches Datenmodell 115

9.6.1 Mapping Quellcode zu EMSG Naming Conventions. 115

9.6.2 Physisches Datenmodell 131

10 Anhang 131

10.1 Datenmodell EMSG-Master 131

Abbildungsverzeichnis

Abbildung 1: Logische Darstellung 15

Abbildung 2: Zuordnung der Anwendungsfälle zur Architektur 17

Abbildung 3: Jahresabschluss 20

Abbildung 4: Schematischer Überblick über die grundlegenden Prozesse der Achsenaktualisierung 22

Abbildung 5: Informationsfluss der Achsdaten bei den zugehörigen Prozessen 22

Abbildung 6: Datenaustausch über Basissystem (INTERLIS 2) 🡺 Fachapplikation 24

Abbildung 7: Transfer Daten BS 🡪 FA unter Verwendung der INTERLIS 2 Schnittstelle und dem MISTRA-Fileserver 28

Abbildung 8: Generierung der Daten für Benchmarking 32

Abbildung 9: Auswertung der Daten für Benchmarking 33

Abbildung 10 Aufbau von Tabellen 34

Abbildung 11 0..n Beziehung 34

Abbildung 12 1..n Beziehung 34

Abbildung 13 0:1 Beziehung 35

Abbildung 14: Datenmodell EMSG-Master 35

Abbildung 15: Tabelle Katalog Belastungskategorie 36

Abbildung 16: Tabelle Katalog Massnahmenvorschlag Global 36

Abbildung 17: Tabelle Katalog Massnahmenvorschlag 37

Abbildung 18: Tabelle Katalog Wiederbeschafftungswert Global 37

Abbildung 19: Tabelle Katalog Wiederbeschaffungswert 37

Abbildung 20: Tabelle Katalog Massnahmen 38

Abbildung 21: Tabelle Katalog Gemeinde 38

Abbildung 22: Tabelle Katalog Öffentliche Verkehrsmittel 38

Abbildung 23: Tabelle Mandant 38

Abbildung 24: Tabelle MandantLogo 39

Abbildung 25: Tabelle Mandant Details für Benchmarking 39

Abbildung 26: Tabelle Erfassungsperiode 39

Abbildung 27: Tabelle Schadengruppe 40

Abbildung 28: Tabelle Schadendetail 40

Abbildung 29: Kenngrössen Früherer Jahre 40

Abbildung 30: Tabelle Kenngrössen Früherer Jahre Detail 40

Abbildung 31: Tabelle Benchmarking 41

Abbildung 32: Tabelle Benchmarking Detail 41

Abbildung 33: Tabellen Summarischer Modus 42

Abbildung 34: Tabellen Strassennamen Modus (tabellarisch) 43

Abbildung 35: Mengentechnische Abhängigkeit zwischen den Stützpunkten von Achssegmenten, Strassenabschnitten und Zustandsabschnitten 44

Abbildung 36: Tabellen im GIS Modus 46

Abbildung 37: Tabelle StrassenabschnittGIS 47

Abbildung 38: Tabelle ZustandsabschnittGIS 48

Abbildung 39: Tabelle RealisierteMassnahmeGIS 49

Abbildung 40: Tabelle MassnahmenvorschlagTeilsystemeGIS 50

Abbildung 41: Tabelle KoordinierteMassnahmeGIS 50

Abbildung 42: EMSG-Master GIS-Datenmodell Struktur 54

Abbildung 43: Datenmodell Inspektionsrouten und Check-Outs 55

Abbildung 44: Datenmodell EMSG-Mobile GIS Daten (Achsen – Referenzen – Fachdaten) 57

Abbildung 45: Architekturskizze 60

Abbildung 46: EMSG-Master Schichten 61

Abbildung 47: OpenLayers JavaScript Kartenelement – Beispiel für Visualisierung von Rasterdaten (WMTS) und Vektordaten (WFS) 66

Abbildung 48: Load Verhalten 67

Abbildung 49: Save Verhalten 68

Abbildung 50: EMSG Proxyfunktion 69

Abbildung 51: Ladeverhalten in EMSG.Mobile 71

Abbildung 52: Speicherverhalten 72

Abbildung 53: Konzept Multimonitor 74

Abbildung 54: MVC3 75

Abbildung 55: Beispiel MVC3 76

Abbildung 56: Konzept Funktionales Testen 77

Abbildung 57: Fertigkomponenten Produktion EMSG-Master 82

Abbildung 58: Inhalt Ressource Datei 86

Abbildung 59: Schnittstellen 90

Abbildung 60: Aktivitätsdiagramm Check-Out von Daten für die Bearbeitung am EMSG-Mobile (Download des Datenpaketes am EMSG-Master) 92

Abbildung 61: Aktivitätsdiagramm Check-Out von Daten für die Bearbeitung am EMSG-Mobile (Importieren des Datenpaketes in EMSG-Mobile) 93

Abbildung 62: Aktivitätsdiagramm Check-In von Daten aus dem EMSG-Mobile in die EMSG-Master Anwendung 94

Abbildung 63: Datenflussskizze 95

Abbildung 64: Schematische Abbildung des SecurityProxies für die Einschränkung der Sichtbarkeit von WMS Services 104

Abbildung 65: Einstieg in EMSG-Master 105

Abbildung 66: Mandantenauswahl 107

Abbildung 67: Developer Client 112

Abbildung 68: Erzeugen des Datenbankschema 113

Abbildung 69: Auswahl Datenbank Instance 113

Abbildung 70: Erzeugen von Stammdaten mit Achsen 113

Abbildung 71: Erzeugen von Stammdaten ohne Achsen 113

Abbildung 72: Bearbeiten von Berechtigungen 114

Abbildung 73: Erzeugen von Testdaten 114

Abbildung 74: Erstellen des Deployment Package 115

# Allgemeines

## Begriffe

| Abkürzung | **Bedeutung** |
| --- | --- |
| AIX | Advanced Interactive Executive |
| AJAX | Asynchronous JavaScript and XML |
| ASP | Active Server Pages |
| ASTRA | Bundesamt für Strassen |
| AV | Amtliche Vermessung |
| BIT | Bundesamt für Informatik und Telekommunikation |
| BS | Basissystem |
| CRUD | Create Read Update Delete |
| CSS | Cascading Style Sheets |
| DBF | Database Files |
| DBMS | Datenbankmanagementsystem |
| DS | Datenschicht |
| EL | IT-Entwicklungsleitung |
| EMSG | Erhaltungsmanagement in Siedlungsgebiet |
| EPSG | European Petroleum Survey Group |
| GDAL | Geospatial Data Abstraction Library |
| GIS | Geographisches Informationssystem |
| GPL | Gesamtprojektleitung MISTRA |
| GL | Geschäftslogik |
| HERMES | Handbuch der elektronischen Rechenzentren des Bundes, eine Methode für die Entwicklung von Systemen |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| ID | Identifier |
| INTERLIS | Datenaustauschmechanismus für Landinformationssysteme |
| IIS | Internet Information Services |
| IT | Informationstechnik |
| LCSFC | Land cover surface |
| LDAP | Lightweight Directory Access Protocol |
| LGPL | GNU Lesser General Public License |
| LNNA | Local names and place name |
| LOCPOS | Locality labels |
| LR | Leitung Realisierung |
| MB | Megabyte |
| MISTRA | Management-Informationssystem Strasse und Strassenverkehr |
| MIT | Massachusetts Institute of Technology |
| MVC | Model View Controller |
| MVVM | Model View ViewModel |
| OGR | GDAL Programmbibliothek für Vektordaten |
| O/R | Objektrelational |
| OSNR | Ownership numbers |
| OSS | Open Source Software |
| PC | Personal Computer |
| PDF | Portable Document Format |
| PL | Projektleitung, sofern die Abkürzung einzeln aufgeführt wird, bezieht sich diese auf die Projektleitung EMSG |
| PR | Präsentationsschicht |
| RBBS | Räumliches Basisbezugssystem Nationalstrassen |
| RE | Realisierung bzw. realisierende Firma |
| RESF | Real estate |
| SDO | Spatial Data Option |
| URL | Uniform Resource Locator |
| USB | Universal Serial Bus |
| WGS | World Geodetic System |
| WMS | Web Map Service |
| WMTS | Web Map Tiled Service (gekacheltes WMS) |
| WMS – T | Web Map Service – Time |
| WPF | Windows Presentation Foundation |
| WS | Web Service |
| XAML | Extensible Application Markup Language |

Die Sammlung sämtlicher Abkürzungen und Begriffsdefinitionen für das Projekt MISTRA werden im MISTRA-Glossar [10] geführt.

## Referenzierte Dokumente

| Nr. | **Titel** | **Autor, Version, Datum** |
| --- | --- | --- |
| 1 | LDAP Web-Services 3.0 Anwenderhandbuch | <name>, Version 1.2 vom 29.07.2009 |
| 2 | EMSG Pflichtenheft RE | Version 1.0 vom 28.01.2011 |
| 3 | Präsentationen Technischer Workshop | Präsentation LR/PL und Präsentation <name>, 23.08.2011 |
| 4 | BS Schnittstellen Web Services R3.0 | <name>, <name>, Version 1.9 vom 04.08.2011 |
| 5 | INTERLIS 2 Referenzhandbuch | Eidg. Vermessungsdirektion, Bern  Version 1 Revision 2, März 1999  http://www.interlis.ch/interlis1/docs/Iref\_12d.pdf |
| 6 | MISTRA Systemarchitektur Gesamtsystem | <name>, Version 1.0 vom 30.06.2008 |
| 7 | EMSG Systemanforderungen | <name>, Version 1.5.0.0 vom 15.12.2014 |
| 8 | EMSG Konzept Multimonitorbetrieb | <name>, <name>, Version 0.3 vom 10.10.2011 |
| 9 | EMSG Usability Konzept | <name>, Version 1.4.0.1 vom 30.06.2014 |
| 10 | MISTRA Glossar | ASTRA, Version 4.3 vom 09.05.2007 |
| 11 | EMSG Konfigurationshandbuch web.config | <name>, Version 1.5.0.0 |
| 12 | EMSG Betriebshandbuch | <name>, Version 1.5.0.0 vom 15.12.2014 |
| 13 | Datenmodell EMSG-Master (EMSG.Architektur.ERM.vsd) | <name>, Version 0.75 vom 28.06.2013 |
| 14 | EMSG Testprotokoll Iteration 3 + Änderungsantrag 04 | <name>, Version 0.17 vom 06.05.2013 |

## Management Summary

Die Fachapplikation EMSG besteht aus folgenden zwei Anwendungen:

1. **EMSG-Master**: Die Hauptapplikation, eine Web-Anwendung für die Verwendung am Büroarbeitsplatz
2. **EMSG-Mobile**: eine Rich Client-Anwendung, die auf mobilen Endgeräten (Tablet PCs) eingesetzt werden kann

Für die Umsetzung von beiden Anwendungen wird das .NET Framework 4.5, unter Verwendung von C# als *Programmiersprache eingesetzt. Sowohl EMSG-Master als auch EMSG-Mobile werden in einer 3-Schichten-Architektur, bestehend aus Präsentations-, Business Logik - und Datenzugriffs-Schicht implementiert.*

Für die Datenhaltung am EMSG-Master wird MS SQL Server 2014 eingesetzt. Als GIS-Server für EMSG-Master wird der GeoServer 2.9.1 verwendet.

Die EMSG-Mobile Rich Client-Anwendung baut auf OpenLayers gehostet in einer WPF-WebBrowser-Komponente auf. GIS und Fachdaten werden bei EMSG-Mobile binär serialisiert gespeichert. Rasterdaten werden in Form von lokalen Karten Caches vorgehalten.

# Zweck des Dokuments

Die **Systemarchitektur** gliedert das System (EMSG) in Subsysteme und Konfigurationseinheiten. Die Systemstruktur und Schnittstellen werden beschrieben. Die Systemarchitektur bildet zusammen mit den dokumentierten Fertigprodukten den Kern der technischen und organisatorischen Umsetzung der Systemanforderungen. Die Systemarchitektur wird vor der Realisierung festgelegt.

Das **Systemdesign** ist eine Verfeinerung der Systemarchitektur und beschreibt die detaillierten Spezifikationen als Vorgabe für die Realisierung des Systems. Das Systemdesign wird in den Iterationen der Realisierung verfeinert.

## Aufgabenstellung

Für die allgemeine Aufgabenstellung des Systems EMSG (Ausgangslage sowie funktionale und nicht-funktionale Ziele) wird auf das EMSG Pflichtenheft RE [2] verwiesen.

Die im EMSG Pflichtenheft RE [2] vorgeschlagene Architektur (EMSG System bestehend auf EMSG- Master und EMSG-Mobile sowie Anbindung an MISTRA Basissystem) bildete die erste Grundlage für dieses Dokument und wird hier verfeinert.

## Architekturziele

Im Folgenden werden die wichtigsten Ziele und Anforderungen, die an die Architektur gestellt werden, beschrieben.

* Das System muss mandantenfähig sein
* Das System soll vom Benutzer einfach zu bedienen sein
* Das System soll möglichst performant sein
* Die Architektur soll „state of the art“ sein
* Die Architektur soll möglichst gut strukturiert und einfach erweiterbar sein
* Das System muss Mehrsprachigkeit unterstützen
* In manchen Bereichen können im EMSG-Master Inhalte in separate Browserfenster ausgelagert werden, um damit mehrere Monitore nutzen zu können.

## Rahmenbedingungen der Architektur

Im EMSG-Master kann der Endanwender vordefinierte Reports (Microsoft Reporting Service) ausführen und das Ergebnis auf seinen Client herunterladenRahmenbedingungen der Architektur

Folgendes beschreibt aus technischer Sicht die Rahmenbedienungen, die an die Architektur gestellt sind:

* EMSG-Master wird auf dem Applikationsserver im IIS in einem eigenen Application Pool mit dem Microsoft .NET Framework 4.5 oder in einem Azure AppService gehostet.
* EMSG-Mobile wird auf einen Windows Tablet mit Microsoft .NET Framework 4.0 betrieben.
* Als GIS-Server wird das Open Source Produkt GeoServer 2.9.2 für den EMSG-Master eingesetzt.

# Lösungsvorschläge für die Struktur des Systems

Für EMSG-Master wird eine webbasierte Lösung (mit Microsoft ASP.NET / MVC3) realisiert. Dadurch entfallen Deployments und Updates auf dem Client des Endanwenders. Deployments werden somit rein in der Produktionsumgebung durchgeführt. Ausserdem muss der Endanwender keine zusätzlichen Plug-Ins im Browser (z.B. für Microsoft Silverlight) installieren.

Der Einstieg in EMSG-Master erfolgt durch den Endanwender durch dessen Browser auf seinem Client. Kartendienste werden über einen konfigurierbaren Direktzugriff des Clients auf die Kartenquellen realisiert, hierbei kann auch noch konfigurativ zwischen WMTS und WMS gewechselt werden sowie ob der Zugriff doch über den Applikationsserver laufen soll. Die EMSG-Instanzdaten (Achsen, Strassenabschnitte usw.) werden immer über einen Serverzugriff (EMSG.Master) und WMS abgebildet.

Für EMSG-Mobile wird eine „Smart Client“-Lösung (mit Microsoft WPF) realisiert. Die Mobile Lösung unterstützt den Offline-Betrieb und bietet mit WPF (Windows Presentation Foundation) eine moderne Benutzeroberfläche an.

# Struktur des Systems

Dieser Abschnitt zeigt und beschreibt den Aufbau der technischen und organisatorischen Aspekte der Subsysteme und Konfigurationseinheiten.

## Logische Sicht

Abbildung 1 zeigt die logische Sicht auf EMSG und stellt die Grundlage dieses Dokuments dar.

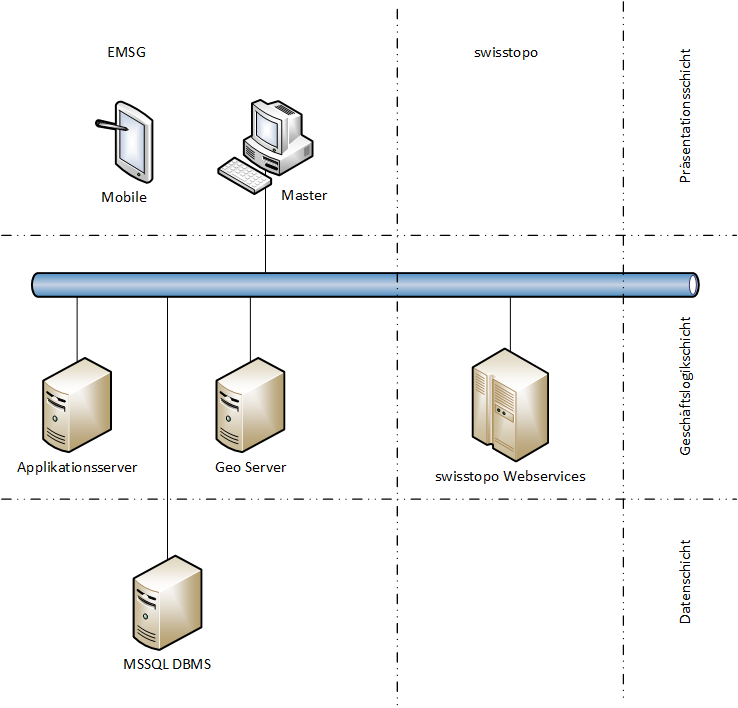


Abbildung 1: Logische Darstellung

### Präsentationsschicht (Clients)

In dieser Ebene befinden sich der Master und der Mobile:

* **EMSG-Master**: Es handelt sich um eine Web-Anwendung mit HTML/AJAX-Frontend, die in ASP.NET MVC3 realisiert wird. Für die Anwendung EMSG-Master wird der Internet Explorer 7 (oder neuer) benötigt. Damit EMSG-Master einwandfrei funktioniert, muss das Ausführen von JavaScript am Client aktiviert sein.
* **EMSG-Mobile**: Es handelt sich um eine Rich Client-Anwendung für Tablet PCs, die mit Windows Presentation Foundation (WPF) realisiert wird. Für die Anwendung EMSG-Mobile wird auf den mobilen Endgeräten die Microsoft .NET 4.0 Runtime benötigt.

### Geschäftslogikschicht

In der mittleren Schicht gibt es in EMSG folgende Server:

* **Applikationsserver**: Als Applikationsserver für EMSG-Master kommt Microsoft IIS (Internet Information Service) in der Version 7 auf Windows Server 2008 R2 zum Einsatz.
* **GeoServer**:Stellt wesentliche GIS-Funktionalitäten, primär für den Bezug von GeoDaten zur Verfügung. Des Weiteren erfolgt die Generierung von Karten im PDF-Format am GeoServer.

Ausserhalb von EMSG werden folgende bereits vorhandene Server/Schnittstellen genutzt:

* **swisstopo**: Betreibt Web Map Services (WMS) und Web Map Tile Services (WMTS) für den Bezug von georeferenzierten Rasterdaten. Auf diese Services wird vom EMSG-Master zwecks Darstellung von Hintergrund- und AV-Karten zugegriffen.

### Datenschicht

* **MSSQL DBMS**:Die Datenhaltung im Master erfolgt mittels MSSQL.
* **EMSG-Mobile**:
  + **Vektordaten**: Werden in Form von binär serialiserten Objekten vorgehalten. Diese erlauben es Fachdaten[[1]](#footnote-2), sowie deren räumliche Definition (Geometrie) zu verwalten.
  + **Rasterdaten**: Werden in Form von Karten-Caches gespeichert. Eine Bearbeitung, bzw. Manipulation der Rasterdaten erfolgt nicht auf EMSG-Mobile.

## Anwendungsfälle-Sicht

Abbildung 2 veranschaulicht, welche Anwendungsfälle durch welchen Client unterstützt werden:

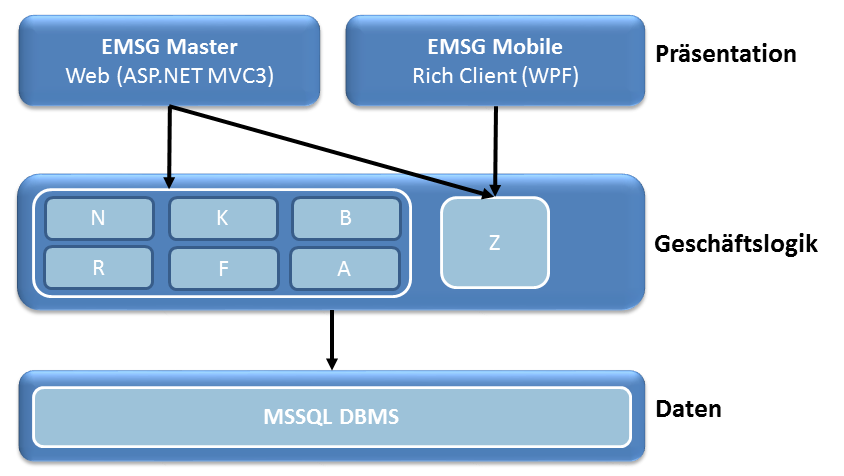


Abbildung 2: Zuordnung der Anwendungsfälle zur Architektur

Die Anwendungsfälle wurden dabei in folgende Module aufgeteilt:

* **N**: Netzverwaltung
* **K**: Massnahmen koordinieren
* **B**: Benchmark
* **R**: Realisierte Massnahmen erfassen
* **F**: Fortschreibung
* **A**: Administration
* **Z**: Zustände und Massnahmen

Details zu den Anwendungsfällen finden sich in [7].

Wie man in Abbildung 2 erkennen kann, stellt der Web-Client (EMSG-Master) die gesamte Funktionalität zur Verfügung. EMSG-Mobile hingegen unterstützt nur folgende Anwendungsfälle im Modul Z (Zustände und Massnahmenvorschläge):

* **Z3** (Schäden und Massnahmenvorschläge für Strassenabschnitte bearbeiten): der Hauptanwendungsfall für den mobilen Einsatz
* **Z5 und Z6**: Für den Import bzw. Export von Daten zwischen den beiden Clients

Ausserdem gibt es im Modul A (Administration) noch das Hilfe-System (A11), das streng genommen kein Anwendungsfall ist (sondern die anderen Anwendungsfälle unterstützt). Dieses steht natürlich auf beiden Clients zur Verfügung.

Es wird versucht, die Geschäftslogik für die Anwendungsfälle so gut wie möglich zwischen EMSG-Master und EMSG-Mobile wiederzuverwenden[[2]](#footnote-3). Die entsprechenden Module werden bei EMSG-Mobile am Client installiert, während sie bei EMSG-Master am Applikationsserver laufen.

## Prozess-Sicht

Dieser Abschnitt beschreibt die aus Sicht der Architektur kritischen Prozesse von EMSG.

### Jahresabschluss in EMSG-Master

Im Idealfall wird jährlich für jeweils das vergangene Kalenderjahr ein Jahresabschluss durchgeführt.

Mit dem Jahresabschluss erfolgt auch die Historisierung der erfassten Daten. Der Jahresabschluss wird am Ende einer Erfassungsperiode (siehe 4.4.4.3) durchgeführt. Eine Erfassungsperiode wird durch einen Datensatz repräsentiert. Erfassungsperioden sind Mandanten zugeordnet. Ein Mandant besitzt immer genau eine aktuelle Erfassungsperiode. Die aktuelle Erfassungsperiode hat noch keinen Jahresabschluss.

Wenn der Benutzer mit EMSG arbeitet, sieht er auf der Benutzeroberfläche immer nur die Fachdaten, die der aktuellen Erfassungsperiode zugeordnet sind. Diese Fachdaten kann der Benutzer in EMSG bearbeiten bzw. neue Fachdaten hinzufügen.

Wird der Jahresabschluss durchgeführt, so wird dies in der aktuellen Erfassungsperiode vermerkt (Erfassungsperiode wird als abgeschlossen gekennzeichnet). Gleichzeitig wird für den Mandanten eine neue Erfassungsperiode erzeugt. Diese ist von nun an für ihn die aktuelle Erfassungsperiode.

Wird der Jahresabschluss im summarischen Modus, Strassennamenmodus (tabellarisch) oder GIS-Modus gemacht, so werden die Netzinformationen vom gewählten Modus kopiert. Diese Kopien werden der neuen Erfassungsperiode zugeordnet.

Die konfigurierbaren Katalogeinträge (Realisierte und koordinierte Massnahmen, Wiederbeschaffungswert, Massnahmenvorschläge) werden – sofern zutreffend – für die neue Erfassungsperiode kopiert und zugeordnet. Änderungen in diesen Bereichen haben somit keine Auswirkung auf abgeschlossene Erfassungsperioden.

Die für das Benchmarking relevanten Mandanten Details (siehe Abbildung 25) werden kopiert und der neuen Erfassungsperiode zugeordnet. Somit ist gewährleistet dass Änderungen in diesem Bereich zu gleichen Ergebnissen im Benchmarking führen.

Aus der Netzinformation werden die notwendigen Informationen berechnet um sie dem Benchmarking zur Verfügung zu stellen. Dieser Schritt ist in 4.3.3.1 detailliert beschrieben.

Die GIS Daten (Geometrien der Achsen, Achssegmente, Strassenabschnitte, Zustandsabschnitte und zugehörige Attribute) werden in jedem Fall kopiert und somit historisiert. Änderungen der Geometrien via Achsen-Update sind immer nur für die aktuelle Erfassungsperiode möglich.

Die Fachdaten von abgeschlossenen Erfassungsperioden können im EMSG nicht weiter bearbeitet werden. Sie stehen ausschliesslich für Auswertungen zur Verfügung. Die Zustandsprotokolle und weitere Informationen der aktuellen und abgeschlossenen Erfassungsperioden können über eine eigene Auswertung abgerufen werden.

Änderungen der Fachdaten der aktuellen Erfassungsperiode haben keine Auswirkung auf Fachdaten voriger Erfassungsperioden.

Fachdaten einer Erfassungsperiode, die nicht dem Abschlussmodus beim Jahresabschluss zugeordnet sind, werden bei der Durchführung des Jahresabschluss gelöscht.

Historisiert werden nur jene Fachdaten, die einer Erfassungsperiode direkt oder indirekt zugeordnet sind (siehe dazu ­­­­­Abbildung 14).

Der Jahresabschluss ist ein Prozess, der länger dauert als andere Anwendungsfälle, die täglich verwendet werden. Deshalb werden im ersten Schritt die nötigen Validierungen durchgeführt, um den Benutzer frühzeitig zu informieren, ob der Jahresabschluss bereits durchgeführt werden kann.

Abbildung 3 zeigt graphisch den Ablauf des Jahresabschluss:



Abbildung 3: Jahresabschluss

Dieser gesamte Prozess nachdem der Benutzer den Jahresabschluss anstösst spielt sich synchron in der Geschäftslogik am Web-Server an.

### Aktualisierung der Achsen im EMSG-Master

Die Aktualisierung der Achsen[[3]](#footnote-4) im EMSG-Master aus dem Basissystem stellt einen komplexen Prozess dar, der hier näher beschrieben wird.

Die Aktualisierung der Achsdaten, die im EMSG-Master die geometrische Grundlage darstellen[[4]](#footnote-5), hat das Ziel, diese Daten im EMSG-Master (und in Folge auf EMSG-Mobile) auf einen aktuellen Stand zu bringen, sodass die Durchführung weiterer Anwendungsfälle auf den aktualisierten Achsdaten erfolgen kann. Ausserdem sollen bereits vorhandene Strassenabschnitte der aktiven Inspektionsperiode so weit wie möglich automatisch auf den aktuellen Stand der Achsen aktualisiert werden. Dort wo eine automatische Aktualisierung der Strassenabschnitte nicht möglich ist, werden dem Benutzer diese Fälle visualisiert und eine manuelle Korrektur der Strassenabschnitte ermöglicht.

Die Aktualisierung der Achsen wird in zwei grundlegende und zeitlich unabhängige Prozesse geteilt (siehe Abbildung 4 und Abbildung 5):

Der Prozess **Datenaustausch Basissystem 🡺 EMSG-Master** importiert in regelmässigen Intervallen[[5]](#footnote-6) Achsänderungen über die INTERLIS 2 Schnittstelle und legt die Achsdaten in separaten Tabellen als „Kopie Achsdaten“ im MSSQL DBMS ab[[6]](#footnote-7). Dieser Datenaustausch hat keine direkten Auswirkungen auf die in EMSG produktiv genutzten Daten und somit auch keine Auswirkungen auf sämtliche EMSG Anwendungsfälle. Der Prozess „Datenaustausch Basissystem 🡺 EMSG-Master“ läuft automatisch im Hintergrund in definierten regelmässigen Intervallen[[7]](#footnote-8) und ohne jegliche Benutzerinteraktion. Details zu diesem Prozess werden in Kapitel 4.3.2.4 beschrieben. Ergebnis dieses Prozesses sind aktuelle Achsdaten für das gesamte Bundesgebiet, die separat von den in EMSG produktiv genutzten Achsdaten im MSSQL DBMS gehalten werden und später bei der Achsenaktualisierung verwendet werden (siehe nächster Abschnitt). Es werden stets die Achsänderungen für die gesamte Schweiz geladen und somit ist dieser Prozess unabhängig von Mandanten.

Der zweite wesentliche Prozess, in Folge als **Achsenaktualisierung EMSG-Master** bezeichnet, hat zum Ziel, die aktuellste Version der Achsdaten in den produktiv genutzten Datenbestand von EMSG zu übernehmen. Der Prozess kann nur durch den EMSG-Master Anwender angestossen werden. Dies kann prinzipiell zu jeder Zeit geschehen – die Durchführung ist aber sinnvollerweise unmittelbar nach dem Jahresabschluss bzw. vor dem Beginn der Eingabe von Zustandsabschnitten durchzuführen[[8]](#footnote-9). Der Prozess kann beliebig oft angestossen werden. Die einzige Voraussetzung für die Durchführung der „Achsenaktualisierung EMSG-Master“ ist, dass keine EMSG Daten für den entsprechenden Mandanten ausgecheckt sind[[9]](#footnote-10). Details zum Prozess „Achsenaktualisierung EMSG-Master“ wurden in Kapitel 4.3.2.5 dokumentiert.

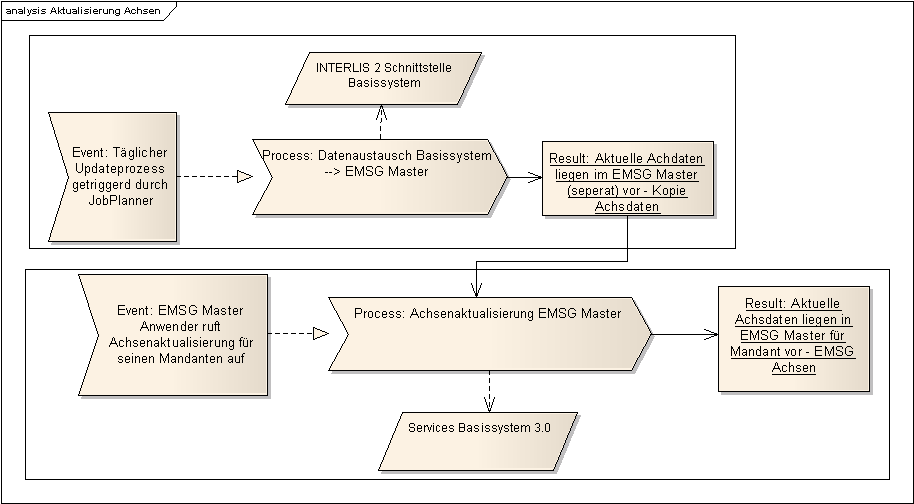


Abbildung 4: Schematischer Überblick über die grundlegenden Prozesse der Achsenaktualisierung

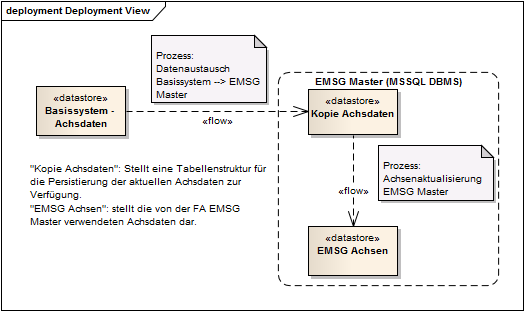


Abbildung 5: Informationsfluss der Achsdaten bei den zugehörigen Prozessen

#### Voraussetzungen

Wie in Abbildung 4 schemenhaft illustriert, nutzen die Prozesse für die Aktualisierung der Achsen im EMSG-Master neben der INTERLIS 2 Schnittstelle auch die Webservices des Basissystems. Folglich werden die folgenden BS-Schnittstellen für die erfolgreiche Durchführung der Achsenaktualisierung in EMSG vorausgesetzt:

* Basissystem: INTERLIS 2 Schnittstelle
* Basissystem: Webservices (BS-WS3.0: LinearReferenceService)

Der Prozess **Datenaustausch Basissystem 🡺 EMSG-Master** verlangt, dass die INTERLIS 2 Schnittstelle in den angegeben regelmässigen Intervallen die entsprechenden Daten zur Verfügung stellt (d.h. die Export-Datei wird im Transferverzeichnis von EMSG abgelegt). Werden keine Daten durch das Basissystem zu Verfügung gestellt, bzw. wenn diese nicht dem INTERLIS2 Datenschema entsprechen, werden diese nicht in das MSSQL DBMS von EMSG-Master übernommen.

#### Transaktionsverhalten

Für beide in Kapitel 4.3.2 skizzierten Prozesse gilt, dass Schreibvorgänge innerhalb einer Transaktion durchgeführt werden.[[10]](#footnote-11) Treten schwerwiegende[[11]](#footnote-12) Fehler in der EMSG-Geschäftslogik oder im MSSQL DBMS während der Durchführung auf, so wird ein Rollback von allen bereits getätigten Änderungen durchgeführt. Damit ist die Datenkonsistenz auch im Fehlerfalle gewährleistet.

#### Versionierung der Achsdaten

Da Achsdaten prinzipiell beliebig oft während eines Jahres aktualisiert werden können, wird ein eigener Versionierungsansatz benötigt, welcher unabhängig vom Jahresabschluss ist. Es wird hierfür eine fortlaufende Import-Nummer verwendet.

#### Datenaustausch Basissystem 🡺 EMSG-Master

Der Ablauf für den Bezug der Achsdaten über die INTERLIS 2 Schnittstelle ist in [6] (Kapitel 5) beschrieben und gilt als Umsetzungsvorlage[[12]](#footnote-13) für die Fachapplikation EMSG. Es werden von der FA EMSG immer nur Achsdaten bezogen und keine Fachdaten an das Basissystem übermittelt. Abbildung 6 zeigt den prinzipiellen Ablauf des Datenaustausches zwischen dem Basissystem und Fachapplikationen.

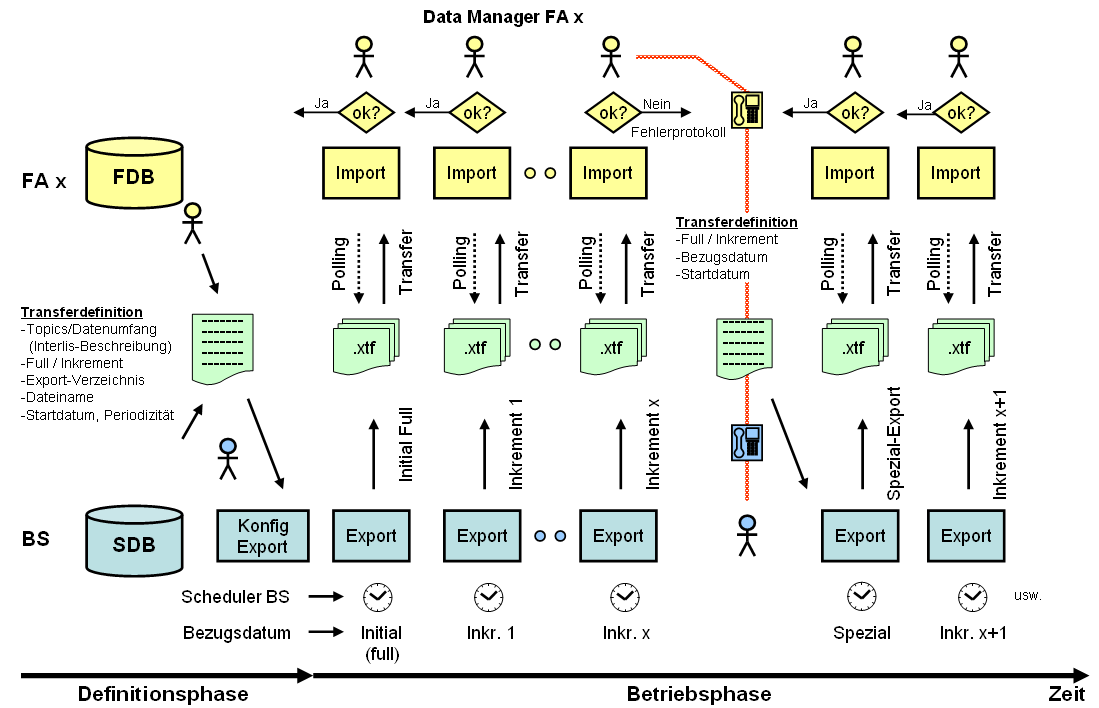


Abbildung 6: Datenaustausch über Basissystem (INTERLIS 2) 🡺 Fachapplikation[[13]](#footnote-14)

Der Prozess „Datenaustausch Basissystem 🡺 EMSG-Master“ unterstützt initiale und

inkrementelle Importe. Der Datenaustausch wird durch die Definition und Übertragung der Transferdefinition[[14]](#footnote-15) gestartet und erfolgt dann inkrementell in den definierten Intervallen. Wesentlich ist, dass die Achsdaten bei EMSG nicht direkt in das produktive Datenmodell übernommen werden, sondern in eigenen Tabellen als „Kopie Achsdaten“ gespeichert werden. Das Ziel dieses Prozesses ist es, eine aktuelle Kopie[[15]](#footnote-16) der Achsdaten für das gesamte Bundesgebiet im MSSQL DBMS vorzuhalten, um in Folge eine effiziente Abwicklung des Prozesses „Achsenaktualisierung EMSG-Master“ zu gewährleisten (siehe Abschnitt 4.3.2.5). Würde in EMSG nicht die aktuellste Version der Achsdaten zur Verfügung stehen, müsste der Benutzer beim Auslösen der Achsenaktualisierung zuerst abwarten, bis die Daten vom BS exportiert und in EMSG importiert werden, was gemäss dem heutigen Stand einen bis mehrere Tage dauern kann (eine spezifische Datenanfrage an BS kann z.Z. nicht automatisiert werden). Durch die automatischen Updates kann EMSG die neusten Achsendaten sofort für die Achsenaktualisierung bereitstellen.

Abbildung 7 illustriert den prinzipiellen Prozessworkflow nach Übermittlung der Transferdefinition, so wie er auch für den Prozess „Datenaustausch Basissystem 🡺 EMSG-Master“ genutzt werden soll. Im Wesentlichen besteht der Prozess aus einem Scheduler[[16]](#footnote-17), der regelmässig am MISTRA-Fileserver prüft, ob Daten im entsprechenden Transferverzeichnis vorhanden sind. Ist dies der Fall, so wird der Importprozess gestartet. Nach erfolgreichem Import der Daten in die Tabellen der „Kopie Achsdaten“ werden durch die Scheduler-Geschäftslogik die entsprechenden INTERLIS 2 Dateien am MISTRA-Fileserver in ein separates Verzeichnis für bereits verarbeitete Daten verschoben und ein entsprechendes Logfile abgelegt.

Tabelle 1 : Die Verzeichnis- und Dateinamensstruktur für den Bezug der Achsdaten entspricht der „**Namenskonvention der Dateien für den INTERLIS-Datenaustausch“.**

|  |  |  |
| --- | --- | --- |
| **Ordner** | **Namenskonvention für Dateinamen** | **Beschreibung** |
| In | [ZeitstempelSender]\_[Interlis-Schemaname].xtf | Vom Basissystem publizierte Daten (Vollständiger Export und inkrementeller Export) |
| Out | [ZeitstempelSender]\_[Interlis-Schemaname].log | Log-Datei Import in EMSG |
| Save | : [ZeitstempelSender]\_[Interlis-Schemaname].xtf | Nach einem erfolgreichen Import wird die Datei vom „In“-Ordner in den „Save“ Ordner verschoben. |

Die Dateien beinhalten im Dateinamen einen Zeitstempel. Der Update-Prozess arbeitet immer jene Interlis-Datei mit dem niedrigsten Zeitstempel zuerst ab. Dieser Vorgang wird so lange wiederholt, bis alle Dateien erfolgreich abgearbeitet wurden (Also der „In“-Ordner leer ist). Wird der Prozess wegen eines Fehlers oder einer fehlerhaften Datei angehalten werden alle ausstehenden (d.h. noch vorhandenen Importdateien) ignoriert und bis zur Behebung des ursächlichen Fehlers nicht bearbeitet.

Die Achs-Kopien werden in den folgenden Tabellen gespeichert:

KOPIEACHSE

Es werden nicht alle Felder der Interlis-Dateien in den EMSG Kopie Tabellen gesichert. Felder, die in den folgenden Auflistungen nicht vorkommen haben für den EMSG Arbeitsablauf keine Relevanz.

|  |  |  |  |
| --- | --- | --- | --- |
| **Feld** | **Typ** | **Wertebereich** | **Bedeutung** |
| BSID | GUID |  | TID aus BS, hier Primary-Key |
| VersionValidFrom | DateTime |  | Achsen-Validity: Beginn Gültigkeit dieser Achsenversion |
| Name | String |  | Textueller Name |
| Owner | String |  | Mandant als Zahlen-/Zeichencode |
| Operation | Int | 2 (Create),  1 (Update), 0 (Delete) | Operation aus INTERLIS |
| ImpNr | Long | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

KOPIEACHSENSEGMENT

|  |  |  |  |
| --- | --- | --- | --- |
| **Feld** | **Typ** | **Wertebereich** | **Bedeutung** |
| BSID | GUID |  | TID aus BS, hier Primary Key |
| AchsenId | GUID |  | Referenz Achse (BSID) |
| Name | String |  | Textueller Name |
| Sequence | String |  | Sequenz-Nummer |
| Operation | Int | 2 (Create),  1 (Update), 0 (Delete) | Operation aus INTERLIS |
| Shape | Geometry (4D Linie) |  | Geometrie |
| ImpNr | Long | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

KOPIESEKTOR

|  |  |  |  |
| --- | --- | --- | --- |
| **Feld** | **Typ** | **Wertebereich** | **Bedeutung** |
| BSID | GUID |  | TID aus BS, hier Primary Key |
| SegmentId | GUID |  | Referenz auf das Achsensegment |
| Km | Double |  | Längswert des Bezugspunktes |
| SectorLength | Double |  | Länge |
| Name | String |  | Textueller Name |
| Sequence | String |  | Sequenz-Nummer |
| Operation | Int | 2 (Create),  1 (Update), 0 (Delete) | Operation aus INTERLIS |
| MarkerGeom | Geometry (Point) |  | Punkt-Geometrie des Sektors |
| ImpNr | Long | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

Zusätzlich wird eine ACHSIMPORTLOG Tabelle geschrieben. In dieser werden die Importvorgänge inkl. Dateipfad und fortlaufender Nummer aufgezeichnet (D.h. pro Importvorgang wird eine Nummer vergeben)

|  |  |  |  |
| --- | --- | --- | --- |
| **Feld** | **Typ** | **Wertebereich** | **Bedeutung** |
| ImpNr | Int | 1,2,3,… | Fortlaufende Import-Nummer: in dieser Tabelle Primärschlüssel |
| Path | String |  | Dateipfad und Name der INTERLIS-Datei im „In“-Verzeichnis |
| Progress | Int | 0 Fehlgeschlagen  1 In Progress, 2 Abgeschlossen | In der Tabelle darf nur ein Datensatz mit dem Wert 1 vorkommen. Ist dies der Fall, müssen alle manuellen Achsupdates der Mandanten gesperrt sein. |
| Timestamp | DateTime |  | Zeitpunkt des Imports (d.h. Empfänger- Zeitstempel) |

##### Import der Datensätze

Die Interlis-Datei wird mit einem XmlReader sequenziell durchlaufen. Stösst der Parser auf eine Definition einer Achse, eines Achssegments oder eines Sektors, wird ein Datensatz in die entsprechende Tabelle geschrieben, oder ein bestehender Datensatz aktualisiert (wenn der eingehende Datensatz mit der Operation ‚UPDATE‘ oder ‚DELETE‘ markiert ist).

Wesentlich ist, dass das Operation-Feld[[17]](#footnote-18) aus Interlis in die Kopie-Tabellen übernommen wird.

Zusätzlich erhalten die Datensätze eine aufsteigende Import-Nummer (ImpNr), die für den späteren Versionsvergleich mit den Produktivdaten verwendet wird.

Die Kopie-Tabellen enthalten immer den Letztstand aus dem Basissystem, inklusive Lösch-Operationen – Damit besteht für alle jemals bekannten Objekte ein Eintrag.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Objekt** | **ImportNr 1** | **ImportNr 2** | **ImportNr 3** | **Aktueller Inhalt Kopie-Tabelle mit Import-Nummer** |
| A | INSERT |  | DELETE | DELETE / 3 |
| B | INSERT | UPDATE | DELETE | DELETE / 3 |
| C | INSERT | UPDATE |  | UPDATE / 2 |
| D | INSERT |  |  | INSERT / 1 |
| E |  | INSERT |  | INSERT / 2 |

Tabelle 2: Inhalt der Achskopie-Tabellen nach initialem und mehrfachem inkrementellen Import

Die Imports erfolgen in einer Transaktion.



Abbildung 7: Transfer Daten BS 🡪 FA unter Verwendung der INTERLIS 2 Schnittstelle und dem MISTRA-Fileserver[[18]](#footnote-19)

#### Achsenaktualisierung EMSG-Master

Der Prozess „Achsenaktualisierung EMSG-Master“ hat das Ziel, die produktiv genutzten Achsdaten für einen Mandanten auf den aktuellsten Stand zu bringen und die Strassen- und Zustandsabschnitte so weit wie möglich automatisiert auf diesen aktuellen Stand nachzuziehen. Ausserdem stellt er die notwendigen Informationen bereit, um eine einfache manuelle Korrektur von Strassen- und Zustandsabschnitten zu gewährleisten falls ein automatisches Nachziehen nicht möglich war.

Der Prozess wird vom Benutzer manuell ausgelöst und läuft im Hintergrund. Eine Fortschrittsanzeige steht nicht zur Verfügung. Wurde der Prozess abgeschlossen, so erhält der EMSG-Anwender eine entsprechende Meldung unter der Voraussetzung, dass er die aktuelle Website nicht verlässt. Es ist aber auch möglich zu einem späteren Zeitpunkt zu prüfen, ob der Prozess abgeschlossen wurde. Dem EMSG-Anwender steht hierfür eine Liste mit den Status der letzten Achsenaktualisierungen zur Verfügung. Pro abgeschlossenen Achsenaktualisierungsprozess ist ersichtlich, ob dieser erfolgreich durchgeführt wurde oder ob ein Fehler aufgetreten ist und somit die Achsen nicht aktualisiert wurden (siehe auch Abschnitte 4.3.2.1 und 4.3.2.2).

Ein Achsenupdate dauert abhängig von der Anzahl der Geometrien und Änderungen, sowie Auslastung des Basissystems und der Datenbank zwischen 2 und 20 Minuten. Genaue empirische Schätzungen sind aufgrund der Dynamik der Faktoren nicht verfügbar. Performance Tests und die darin ermittelten Zeiten wurden im Testprotokoll [14] ermittelt.

Wesentlich ist, dass dieser Prozess keine Daten über die INTERLIS 2 Schnittstelle bezieht, sondern die aktuellen Achsen aus dem lokalen im MSSQL DBMS gespeicherten Datenbestand „Kopie Achsdaten“ bezieht (siehe auch Abbildung 5). Diese Daten wurden zuvor durch den Prozess „Datenaustausch Basissystem 🡺 EMSG-Master“ mit der letzten Version aus dem Basissystem synchronisiert.

In diesem Schritt wird auch die Zuordnung der Alphanumerischen Mandanten-IDs auf die Einträge der Mandanten-Tabelle vorgenommen.

Während der Achsenaktualisierung wird die Bearbeitung für den jeweiligen Mandanten gesperrt. Der Fortschritt wird in der Tabelle ACHSUPDATELOG gespeichert. Weiter wird verhindert, dass das Achsenupdate mehrmals aufgerufen wird.

Treten bei der Achsenaktualisierung Fehler auf – beispielsweise wenn die Übertragung eines Strassenabschnitts nicht möglich ist – so wird dies durch einen Eintrag in der Tabelle ACHSUPDATECONFLICT vermerkt. Dieser enthält die vorherige Geometrie des referenzierten Achsensegments.

Der Prozess „Achsenaktualisierung EMSG-Master“ wird in Folge in drei Schritte getrennt und detaillierter beschrieben:

1. Detektion der Achsdatenänderungen
2. Automatische Aktualisierung der Strassenabschnitte im EMSG-Master
3. Aufbereitung Visualisierung für manuelle Korrektur der Strassenabschnitte

##### Schritt 1: Detektion der Achsdatenänderungen

Die Detektion der Achsdatenänderungen erfolgt über den Vergleich der Einträge der Achsen-Kopie Tabellen und den produktiv genutzten Achsendaten des jeweiligen Mandaten. Der Vergleich erfolgt über die Felder „BSID“ (Objekt-ID-Basissystem) und „ImpNr“ (Import-Nummer). Alle Datensätze, die in den Kopie-Achsdaten-Tabellen eine höhere „ImpNr“ zugewiesen haben, müssen nachgezogen werden. Datensätze, die in den produktiv genutzten Achsentabellen des Mandaten noch nicht vorhanden sind, müssen dort neu eingespielt werden.

Tabelle 3: Funktionsweise Achsen-Aktualisierung

|  |  |  |  |
| --- | --- | --- | --- |
| **Achskopie, Wert des Operation-Feldes** | **Achse produktiv** | **Aktion Achse produktiv** | **Aktualisierung referenzierende Objekte (Strassenabschnitt, Zustand)** |
| INSERT | Nicht vorhanden | INSERT | Keine, aber Warnung[[19]](#footnote-20) |
| UPDATE | Nicht vorhanden | INSERT | Keine[[20]](#footnote-21) |
| UPDATE | Vorhanden | UPDATE | JA (UpdateLinearStable) |
| DELETE | Nicht vorhanden | Keine | Keine |
| DELETE | Vorhanden | DELETE | Ja (Löschung)[[21]](#footnote-22) |

##### Schritt 2: Automatische Aktualisierung der Strassen- und Zustandsabschnitte im EMSG-Master

Strassen- und Zustandsabschnitte in der FA EMSG leiten sich geometrisch aus den Geometrien der Achssegmente ab und verweisen über Achsen-Referenzen indirekt auf diese. Da Strassenabschnitte eine stabile Entität über die Zeit darstellen ist es notwendig, dass diese im Zuge der Aktualisierung der Achsen nachgezogen werden. Die EMSG-Master Geschäftslogik versucht, bestehende Strassen- und Zustandsabschnitte automatisch an die neuen Achsgeometrien anzupassen. In Fällen, wo eine automatische Anpassung der Strassen- und Zustandsabschnitte nicht möglich ist, muss diese in Folge manuell vom Benutzer (siehe Schritt 3) angepasst werden.

##### Schritt 3: Aufbereitung Visualisierung für manuelle Korrektur der Strassen- und Zustandsabschnitte

Um eine einfache manuelle Nachbearbeitung der Strassen- und Zustandsabschnitte zu erlauben, werden im EMSG-Master Änderungen der Achsen visualisiert. Hierfür wird ein eigener GIS-Layer erzeugt, der folgenden Inhalt aufweist:

* Geometrien der Achssegemente, bei welchen eine Übertragung referenzierender Objekte (Strassenabschnitte oder Zustände) auf die neueren Version nicht möglich war.

Diese „Fehlergeometrien“ können als eigener Karten-Layer ein- und ausgeblendet werden, sie werden erst beim nächsten Achsenupdate aus der Datenbank entfernt um neue Konflikte darstellen zu können.

### Benchmarking

Die Daten für Benchmarking werden beim Jahresabschluss (siehe 4.3.1) generiert und dem Benchmarking zur Verfügung gestellt. Die Daten werden in eigenen Tabellen aus Performancegründen abgelegt und werden beim Jahresabschluss bereits berechnet sodass sie später nur mehr gelesen werden müssen.

#### Generierung der Daten

Die Daten für das Benchmarking werden beim Jahresabschluss erzeugt. Jeder Mandant stellt seine erfassten Fachdaten immer für das Benchmarking zur Verfügung. Unabhängig davon welche Rollen der Benutzer bei der Durchführung des Jahresabschluss besitzt.

Im ersten Schritt werden die Netzinformationen verarbeitet. Im tabellarischen sowie im GIS Modus werden nur Netzinformationen von Gemeinden herangezogen.

Im zweiten Schritt werden die Realisierten Massnahmen verarbeitet. Diese werden uneingeschränkt in allen Modi verarbeitet.

Im letzten Schritt werden die transformierten Daten in eigenen Tabellen (siehe 4.4.4.7) gespeichert und stehen ab diesen Zeitpunkt dem Benchmarking (siehe 4.3.3.2) zur Verfügung.

Die Generierung der Daten wird in Abbildung 8 veranschaulicht.



Abbildung 8: Generierung der Daten für Benchmarking

#### Auswertung der Daten

Die Benchmarking Daten stehen nur Benutzern zur Verfügung welche die Rolle Benchmarkteilnehmer auf ihren Mandanten besitzen (siehe Zuordnung Rollen zu Anwendungsfällen in Tabelle 10).

Das Benchmarking kann ein Benutzer immer nur für ein Bezugsjahr generieren. Wenn der Benutzer im Rahmen des Benchmarking eine Auswertung aufruft so wird ihm im ersten Schritt eine Liste der Bezugsjahre präsentiert. Bei den Bezugsjahren handelt es sich dabei um eine Liste von Kalenderjahren wobei jedes Kalenderjahr einer Erfassungsperiode des Mandanten entspricht.

Im Anschluss ermittelt das System jene Mandanten mit welchen der Mandanten des Benutzers verglichen werden kann. Wurden genügend Mandanten zum Vergleich gefunden werden die Daten des Benchmarking mit den Kenngrössen früherer Jahre für das entsprechende Bezugsjahr aufbereitet und den Auswertungen zur Verfügung gestellt.

Der Auswertung der Daten wird in Abbildung 9 veranschaulicht.



Abbildung 9: Auswertung der Daten für Benchmarking

## Daten-Sicht

Es wird ein relationales Datenmodell bis zur dritten Normalform eingesetzt. In dem Datenmodell werden alle Fachdaten und GIS-spezifischen Informationen von EMSG gespeichert.

### Aufbau von Tabellen im Datenmodell

Abbildung 10 zeigt schematisch wie Tabellen im Datenmodell beschrieben.



Abbildung 10 Aufbau von Tabellen

### Kardinalität

Im Folgenden werden kurz die unterschiedlichen Kardinalitäten erwähnt welche später in dem Datenmodell zum Einsatz kommen. Eingesetzt wird die Martin-Notation[[22]](#footnote-23) welche auch als *crow’s feet notation* bekannt ist.

#### 0..n Beziehung



Abbildung 11 0..n Beziehung

#### 1..n Beziehung



Abbildung 12 1..n Beziehung

#### 0:1 Beziehung

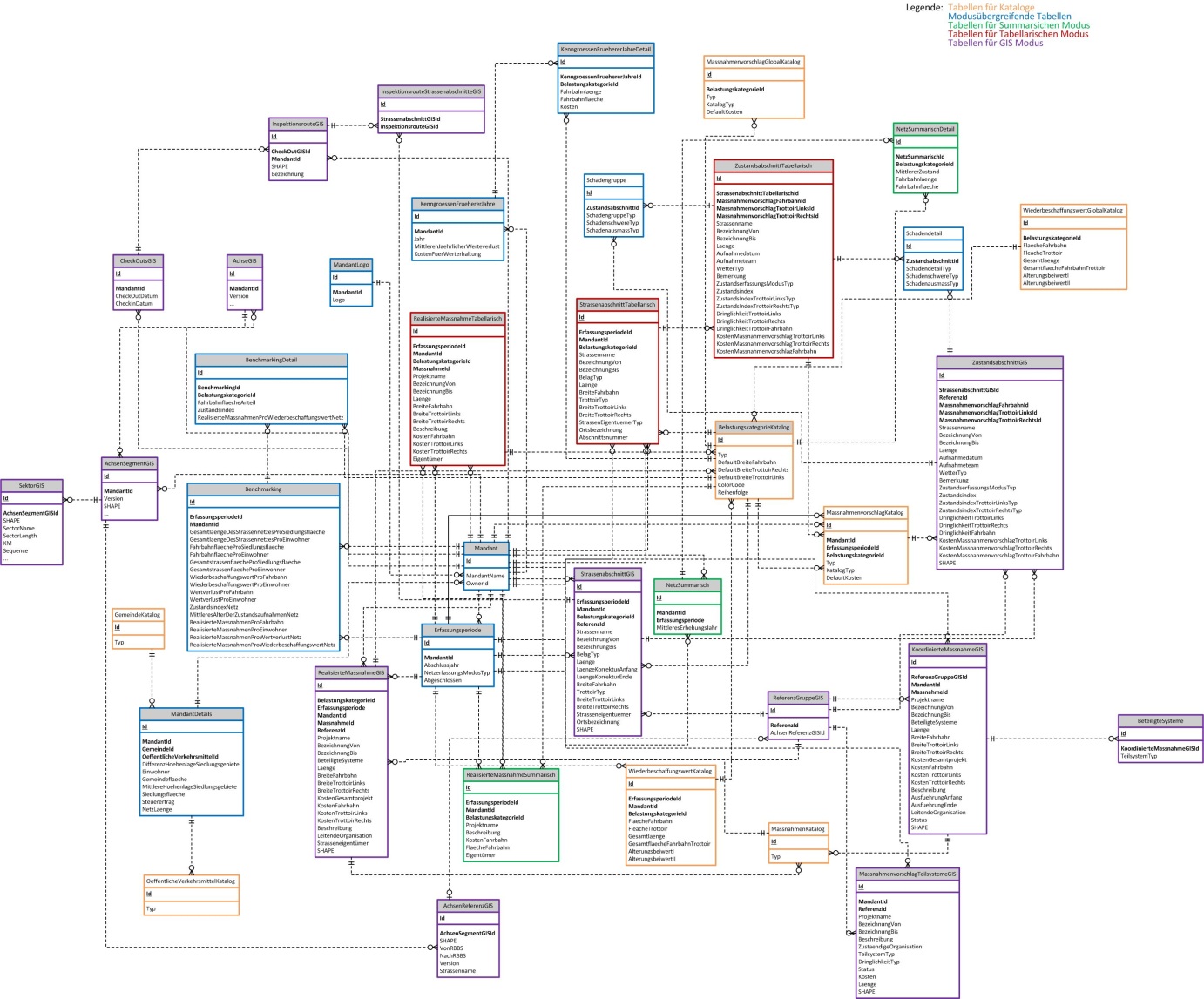


Abbildung 13 0:1 Beziehung

### Namenskonvention

Für die Bezeichnungen der Tabellen und Spalten in den Entitäten wird die Schreibweise CamelCase gewählt. Die Abbildung in der Datenbank erfolgt hingegen gemäss „MISTRA Basissystem, Datenkatalog Sockeldaten“, Kapitel 5 und MISTRA Gesamtarchitektur, Kapitel 7.

Datenmodell



­­­­­Abbildung 14: Datenmodell EMSG-Master

­­­­­Abbildung 14 zeigt eine Gesamtübersicht alle Tabellen von EMSG-Master[[23]](#footnote-24). In den darauf folgenden Abschnitten werden Ausschnitte davon näher beschrieben.

### Modusübergreifende Tabellen

Im Folgenden werden jene Tabellen vorgestellt, die in allen drei Bearbeitungsmodi (siehe [7]) verwendet werden. Inhaltliche Tabellen sind klar einem Modus zugeordnet. Folgende Suffixe in den Tabellennamen werden für die Zuordnung verwendet:

* **Summarisch**: Eindeutig dem Summarischen Modus zugeordnete Tabellen.
* **Tabellarisch**: Eindeutig dem Strassennamen Modus zugeordnete Tabellen.
* **GIS**: Eindeutig dem GIS-Modus zugeordnete Tabellen.

Tabellen, die nicht der obig aufgeführten Nomenklatur entsprechen, sind allgemeine Tabellen, die zu mindestens in zwei Modi der EMSG Fachapplikation verwendet werden.

#### Kataloge

Alle dynamischen Wertevorräte werden in der Datenbank jeweils in einer eigenen Tabelle abgelegt. Als Suffix für die Tabelle wird **Katalog** verwendet. Bei Verwendung wird in der konkreten Tabelle der Fremdschlüssel auf den Wertevorrat gespeichert.

Alle statischen Wertevorräte werden im Code als eigener *enum* abgebildet. Bei Verwendung in der Datenbank werden die Wertevorräte als *integer* abgebildet. Die Attribute in der Tabelle haben dabei das Suffix **Typ**.

##### Belastungskategorie

Für jede Belastungskategorie existiert ein Datensatz in der Tabelle.

Abbildung 15 zeigt den Aufbau der Tabelle.



Abbildung 15: Tabelle Katalog Belastungskategorie

##### Massnahmenvorschlag Global

Diese Tabelle beinhaltet alle, globalen Massnahmenvorschläge.



Abbildung 16: Tabelle Katalog Massnahmenvorschlag Global

##### Massnahmenvorschlag

Jeder Mandant kann seine eigenen Massnahmenvorschläge innerhalb einer Erfassungsperiode konfigurieren. Jede Konfiguration eines Mandanten entspricht einem Datensatz.



Abbildung 17: Tabelle Katalog Massnahmenvorschlag

##### Wiederbeschaffungswert Global

Diese Tabelle beinhaltet alle, globalen Wiederbeschaffungswerte.



Abbildung 18: Tabelle Katalog Wiederbeschafftungswert Global

##### Wiederbeschaffungswert

Jeder Mandant kann seine eigenen Wiederbeschaffungswerte innerhalb einer Erfassungsperiode konfigurieren. Jede Konfiguration eines Mandanten entspricht einem Datensatz.



Abbildung 19: Tabelle Katalog Wiederbeschaffungswert

##### Massnahmen

Für jedes Massnahme existiert ein Datensatz in der Tabelle.



Abbildung 20: Tabelle Katalog Massnahmen

##### Gemeinde

Für jede Gemeinde existiert ein Datensatz in der Tabelle.



Abbildung 21: Tabelle Katalog Gemeinde

##### ÖffentlicheVerkehrsmittel

Für jedes Öffentliche Verkehrsmittel existiert ein Datensatz in der Tabelle.



Abbildung 22: Tabelle Katalog Öffentliche Verkehrsmittel

#### ­­Mandant

Jeder Mandant wird durch einen Datensatz in der Tabelle repräsentiert. Um in den EMSG-Master als Benutzer für einen Mandanten einsteigen zu können, ist es notwendig, dass der Mandant des Benutzers in dieser Tabelle eingetragen ist. Abbildung 23 zeigt den Aufbau der Tabelle.



Abbildung 23: Tabelle Mandant

Das Logo eines Mandanten wird in einer eigenen Tabelle gespeichert. Das Logo wird nur für die Reports verwendet und wird somit nur im Bedarfsfall geladen.



Abbildung 24: Tabelle MandantLogo

Die Mandanteigenschaften für das Benchmarking werden in einer eigenen Tabelle gespeichert. Die Eigenschaft eines Mandanten werden mit jedem Jahresabschluss historisiert.



Abbildung 25: Tabelle Mandant Details für Benchmarking

#### Erfassungsperiode

Für jeden Mandant wird der Jahresabschluss über einen Eintrag in der Tabelle Erfassungsperiode realisiert. Pro Mandant und Jahr kann es nur einen Eintrag geben. Beim Jahresabschluss wird angegeben mit welchem Erfassungsmodus die Eingabe abgeschlossen wird. Abbildung 26 zeigt den Aufbau der Tabelle.



Abbildung 26: Tabelle Erfassungsperiode

Der Vorgang des Jahresabschluss wurde in Abschnitt 4.3.1 erläutert.

#### Schadengruppe

In der groben Zustandserfassung wird pro Zeile im Formular ein eigener Datensatz in dieser Tabelle gespeichert.



Abbildung 27: Tabelle Schadengruppe

#### Schadendetail

In der detaillierten Zustandserfassung wird pro Zeile im Formular ein eigener Datensatz in dieser Tabelle gespeichert.



Abbildung 28: Tabelle Schadendetail

#### Kenngrössen früherer Jahre

Die „Kenngrössen früherer Jahre“ werden jahresweise gespeichert. In den Details befinden sich die Werte für jede einzelne Belastungskategorie.



Abbildung 29: Kenngrössen Früherer Jahre



Abbildung 30: Tabelle Kenngrössen Früherer Jahre Detail

#### Benchmarking

Beim Jahresabschluss werden die Daten für das Benchmarking ermittelt und für jeden Mandanten gespeichert.



Abbildung 31: Tabelle Benchmarking



Abbildung 32: Tabelle Benchmarking Detail

### Summarischer Modus

Dieser Modus realisiert die Erfassung des Strassennetzes in summarischer Form. Die Ausmasse werden dabei pro Belastungskategorie erfasst. Anwender sind z.B. grössere Gemeinden, die Drittsysteme einsetzen.

Abbildung 33 gibt einen Überblick aller betroffenen Tabellen die im summarischen Modus verwendet werden.

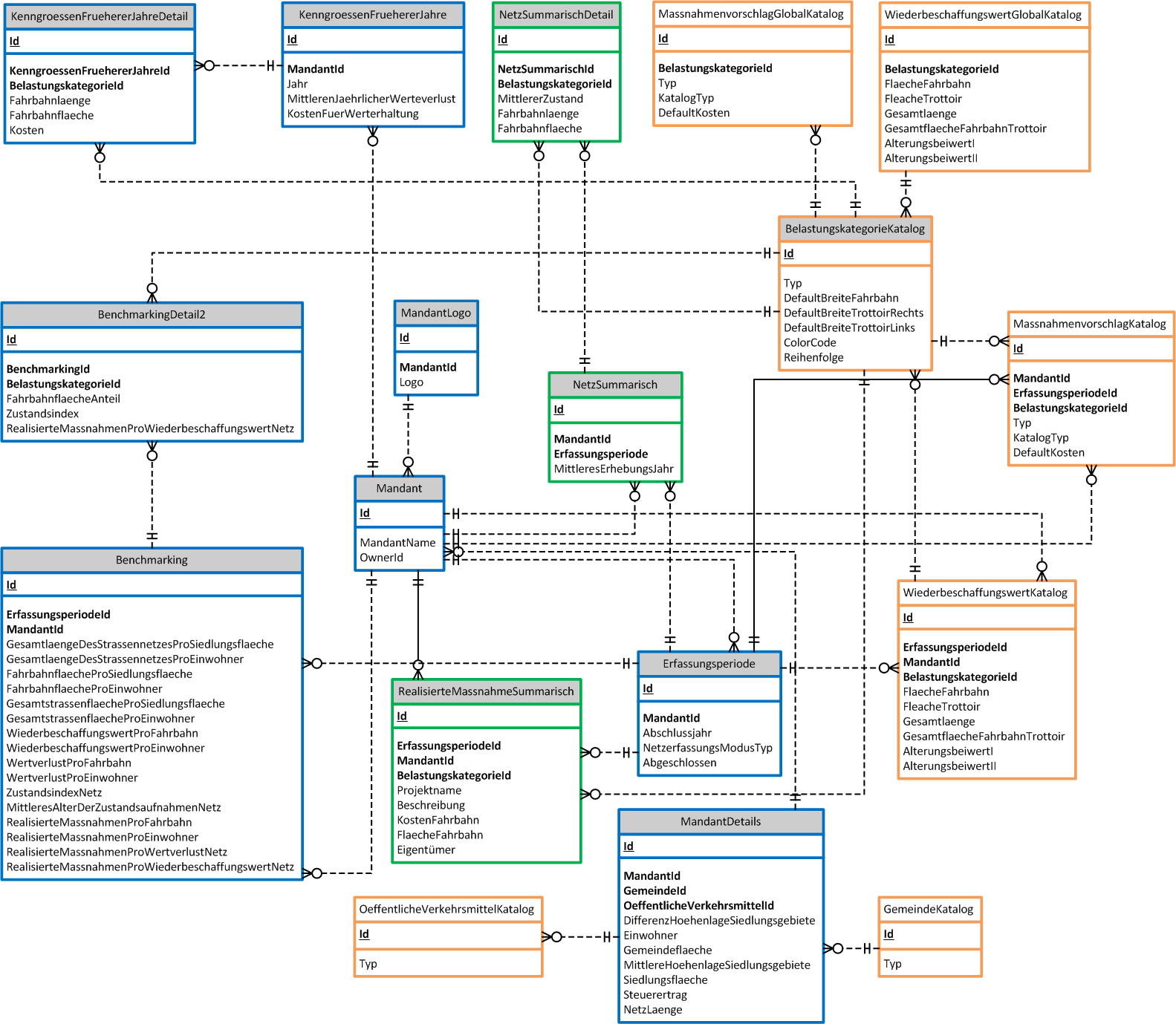


Abbildung 33: Tabellen Summarischer Modus

* **NetzSummarisch**:Für jede Erfassungsperiode gibt es in der Tabelle NetzSummarisch *n* Zeilen.
* **NetzSummarischDetail**: Zu jeder Periode gibt es in der Tabelle NetzSummarischDetail *n* Zeilen. Jede Zeile enthält die Daten für eine konkrete Belastungskategorie.
* **RealisierteMassnahmeSummarisch**: Zu jeder Periode gibt es in der Tabelle RealisierteMassnahmeSummarisch *n* Zeilen.

### Strassennamen Modus (tabellarisch)

Dieser Modus realisiert die Erfassung des Netzes in Form von Strassenabschnitten. Die detaillierte Erfassung von Zuständen und Massnahmenvorschlägen erfolgt für in Bezug auf den Zustand homogene Zustandsabschnitte. Die Basis für die Definition dieser Zustandsabschnitte stellen die Strassenabschnitte dar.

Abbildung 34 gibt einen Überblick über alle Tabellen, die im Strassennamen Modus (tabellarisch) verwendet werden[[24]](#footnote-25).



Abbildung 34: Tabellen Strassennamen Modus (tabellarisch)

* **StrassenabschnittTabellarisch**: Jeder erfasste Strassenabschnitt wird durch einen Datensatz in der Tabelle StrassenabschnittTabellarisch repräsentiert. Es kann beliebig viele Strassenabschnitte pro Erfassungsperiode geben.
* **ZustandsabschnittTabellarisch**: Ein Zustandsabschnitt wird durch einen Eintrag in der Tabelle ZustandsabschnittTabellarisch realisiert. Ein Strassenabschnitt kann 0 bis n Zustandsabschnitte besitzen.
* **RealisierteMassnahmeTabellarisch**: Zu jeder Periode gibt es in der Tabelle RealisierteMassnahmeTabellarisch *n* Zeilen.
* **Schadendetail:** Jeder Zustandsabschnitt besitzt für die Abbildung des Zustandsformulars 0 bis n Schadendetail-Informationen.
* **Schadengruppe:** Jeder Zustandsabschnitt besitzt für die Abbildung des Zustandsformulars 0 bis n Schadengruppe-Informationen.

Die Fahrbahnfläche, Fläche des Trottoir bzw. die Gesamtfläche werden in der Geschäftslogik „on the fly“ berechnet, d.h. nicht in der Datenbank persistiert.

### GIS-Modus

Zusätzlich zu den im Strassennamenmodus erfassten EMSG- Fachdaten werden die Objekte im GIS-Modus um exakte räumliche Definition (Geometrie) erweitert. Zusätzlich zu den fachlich im Strassennamenmodus definierten Objekten werden im GIS-Modus weitere hinzugefügt:

* Inspektionsrouten
* Koordinierte Massnahmen
* Achssegment, Achse und Sektor[[25]](#footnote-26)

#### Geometrien

EMSG persistiert Geometrien im MSSQL DBMS als Datentyp GEOMETRY. Der Zugriff auf die EMSG-Geodaten erfolgt direkt über den GeoServer (Query Layer[[26]](#footnote-27)) und über den O/R-Mapper NHibernate. Die Manipulation der Geodaten erfolgt ausschliesslich über den O/R-Mapper NHibernate nach erfolgreicher Validierung durch die EMSG-Master Geschäftslogik. Hierdurch wird sichergestellt, dass die Datenkonsistenz auch aus räumlicher Sicht durchgängig gewährleistet ist. EMSG generiert keine neuen Geometrien durch eine Anwendereingabe. Alle erzeugten Geometrien stellen eine definierte Teilmenge der Achssegmente (Stützpunkte/Verteces der Achssegmente) dar und sind somit abgleitet bzw. aggregierte Geometrien. Abbildung 35 illustriert die mengentechnische Abhängigkeit zwischen Achssegmenten, Strassenabschnitten und Zustandsabschnitten.



Abbildung 35: Mengentechnische Abhängigkeit zwischen den Stützpunkten von Achssegmenten, Strassenabschnitten und Zustandsabschnitten

Die in Abbildung 35 dokumentiert mengentechnische Abhängigkeit wird in der Datenbank nicht abgebildet (z.B. über entsprechende Check Constraints). Für die Erhaltung dieser Konsistenz ist die EMSG Geschäftslogik zuständig.

#### ****Koordinatensystem & Projektion****

Alle GIS Daten werden in EPSG:21781 (CH1903 / LV03) gespeichert und dargestellt.

* WGS84 Bounds: 5.9700, 45.8300, 10.4900, 47.8100
* Projected Bounds: 485869.5728, 76443.1884, 837076.5648, 299941.7864

|  |
| --- |
| PROJCS["CH1903 / LV03",  GEOGCS["CH1903",  DATUM["CH1903",  SPHEROID["Bessel 1841",6377397.155,299.1528128,  AUTHORITY["EPSG","7004"]],  TOWGS84[674.374,15.056,405.346,0,0,0,0],  AUTHORITY["EPSG","6149"]],  PRIMEM["Greenwich",0,  AUTHORITY["EPSG","8901"]],  UNIT["degree",0.01745329251994328,  AUTHORITY["EPSG","9122"]],  AUTHORITY["EPSG","4149"]],  UNIT["metre",1,  AUTHORITY["EPSG","9001"]],  PROJECTION["Hotine\_Oblique\_Mercator"],  PARAMETER["latitude\_of\_center",46.95240555555556],  PARAMETER["longitude\_of\_center",7.439583333333333],  PARAMETER["azimuth",90],  PARAMETER["rectified\_grid\_angle",90],  PARAMETER["scale\_factor",1],  PARAMETER["false\_easting",600000],  PARAMETER["false\_northing",200000],  AUTHORITY["EPSG","21781"],  AXIS["Y",EAST],  AXIS["X",NORTH]] |

#### Tabellen im GIS-Modus

Abbildung 36 zeigt die Tabellen, in denen die Fachdaten abgelegt werden.

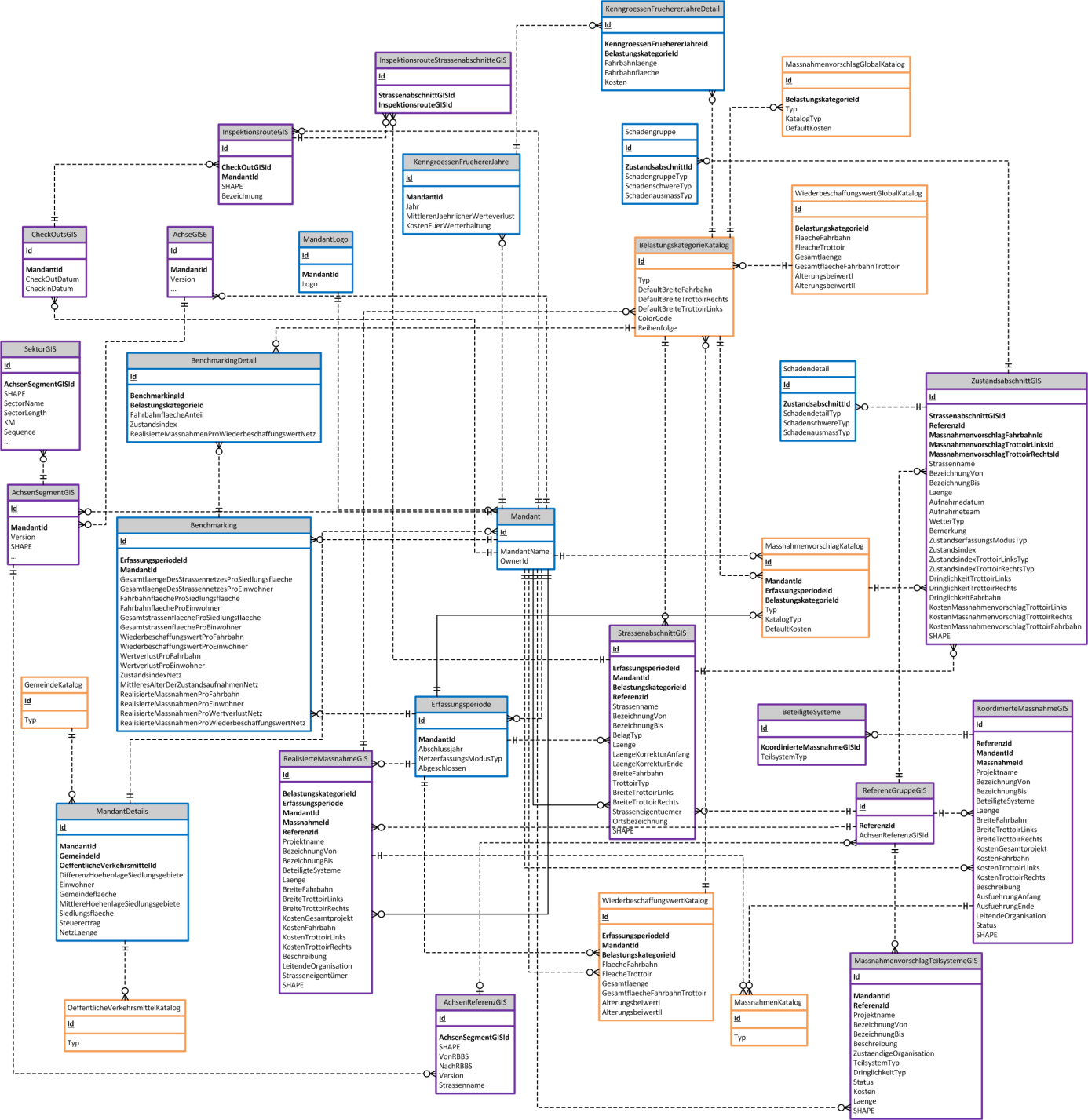


Abbildung 36: Tabellen im GIS Modus

StrassenabschnittGIS



Abbildung 37: Tabelle StrassenabschnittGIS

Jeder erfasste Strassenabschnitt wird durch einen Datensatz in der Tabelle StrassenabschnittGIS repräsentiert. Es kann beliebig viele Strassenabschnitte pro Erfassungsperiode geben. Strassenabschnitte sind Achssegmenten (AchsegmentGIS) mit einer 1:n Assoziation zugeordnet und können sich damit über mehrere, aber auch nur Teile von Achssegmenten beziehen. Ein Strassenabschnitt liegt somit über eine Referenzgruppe auf n Achssegmenten; Allerdings liegt auf einem Achssegment (über eine entsprechende Referenzgruppe) ebenso ein oder mehrere Strassenabschnitte. Aus Sicht eines Straßenabschnittes liegt dennoch eine 1:n Verbindung vor.

ZustandsabschnittGIS



Abbildung 38: Tabelle ZustandsabschnittGIS

Ein Zustandsabschnitt wird durch einen Eintrag in der Tabelle ZustandsabschnittGIS realisiert. Ein Strassenabschnitt kann 0 bis n Zustandsabschnitte besitzen. Zustandsabschnitte sind damit geometrisch eine Teilmenge eines einzelnen Strassenabschnittes. Diese geometrische Relation wird in der Datenschicht nicht durch Constraints geprüft, sondern wird durch die EMSG Geschäftslogikschicht gewährleistet.

RealisierteMassnahmeGIS



Abbildung 39: Tabelle RealisierteMassnahmeGIS

Jede erfasste „realisierte Massnahme“ (RealisierteMassnahmeGIS) wird durch einen Datensatz in der Tabelle RealisierteMassnahmeGIS repräsentiert. Geometrien von realisierten Massnahmen werden auf Basis von Strassenabschnitten (StrassenabschnitteGIS) definiert. Eine realisierte Massnahme kann sich auf mehrere Teile von Strassenabschnitte beziehen und stellt damit eine geometrische Teilmenge von mehreren Strassenabschnitten dar. Diese geometrische Relation wird in der Datenschicht nicht durch Constraints geprüft, sondern wird durch die EMSG Geschäftslogikschicht gewährleistet.

MassnahmenVorschlagTeilsystemeGIS



Abbildung 40: Tabelle MassnahmenvorschlagTeilsystemeGIS

Jeder erfasster Massnahmenvorschlag der Teilsysteme wird in der Tabelle MassnahmenvorschlagTeilsystemeGIS persistiert. Diese verweisen, so wie die realisierten Massnahmen mit einer n:m Relation auf Strassenabschnitte.

KoordinierteMassnahmeGIS



Abbildung 41: Tabelle KoordinierteMassnahmeGIS

Eine koordinierte Massnahme im GIS-Modus wird in der Tabelle KoordinierteMassnahmeGIS persistiert. Deren Geometrie leiten sich wie bei den realisierten Massnahmen aus mehreren Strassenabschnitten ab. Koordinierte Massnahmen werden als fortlaufende Liste geführt und sind vom Jahresabschluss nicht betroffen.

Achsen/Achssegment/Sektor

Achsdaten, also die Achse (AchseGIS), das Achssegment (AchsSegmentGIS) und die zugehörigen Sektoren (SektorGIS) werden in den Tabellen „AchseGIS“, „AchsSegmentGIS“ und „SektorGIS gespeichert. Die Daten werden durch den Prozess „Achsenaktualisierung“ (siehe Abschnitt 4.3.2) verändert, und dienen im EMSG-Master als geometrische Grundlage (siehe Abschnitt 4.4.7.1).

Die Achsdaten werden gemeinsam mit den daraufliegenden Strassenabschnitten und Zustandsabschnitten mit jedem Jahresabschluss dupliziert. Die Achsenaktualisierung erneuert nur die Achsdaten der aktuellen Erfassungsperiode. Damit bleiben die Daten früherer Erfassungsperioden konsistent. Die Duplizierung erleichtert weiters die Darstellungs-Filterung im GeoServer

Folglich kann die ID-Spalte aus dem Basis-System („BSID“) nicht als Primärschlüssel verwendet werden. Es wird eine eigene (EMSG interne) ID-Spalte („ID“) verwendet.

ACHSEGIS

Fehlende Felder aus den Interlis Dateien haben für EMSG keine Relevanz und werden somit nicht übernommen (dazu zählen: Typ, Gültig bis und Status).

|  |  |  |  |
| --- | --- | --- | --- |
| **Feld** | **Typ** | **Wertebereich** | **Bedeutung** |
| ID | GUID |  | Primärschlüssel – Interner EMSG Id. |
| BSID | GUID |  | TID aus BS |
| VersionValidFrom | DateTime |  | Achsen-Validity: Beginn Gültigkeit dieser Achsenversion |
| Name | String |  | Textueller Name |
| MandantID | GUID |  | Zuordnung zu Mandanten-Tabelle |
| ErfassungsperiodID | GUID |  | Erfassungsperiode |
| Operation | Int | 2 (Create),  1 (Update) , 0 Delete | Operation aus Interlis |
| ImpNr | Long | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

ACHSENSEGMENTGIS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feld** | **Typ** | | **Wertebereich** | **Bedeutung** |
| ID | GUID | |  | Primärschlüssel – Interner EMSG Id. |
| BSID | GUID |  | | TID aus BS |
| AchseID | GUID | |  | Referenz Achse |
| MandantID | GUID |  | | Zuordnung zu Mandanten-Tabelle |
| ErfassungsperiodID | GUID |  | | Erfassungsperiode |
| Name | String | |  | Textueller Name |
| Sequence | String | |  | Sequenz-Nummer |
| Operation | Int | | 2 (Create),  1 (Update) , 0 Delete | Operation aus Interlis |
| Shape | Geometry (4D Linie) | |  | Geometrie |
| ImpNr | Long | | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

SEKTORGIS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feld** | **Typ** | | **Wertebereich** | **Bedeutung** |
| ID | GUID | |  | Primärschlüssel – Interner EMSG Id. |
| BSID | GUID |  | | TID aus BS |
| SegmentID | GUID | |  | Referenz auf das Achsensegment |
| MandantID | GUID |  | | Zuordnung zu Mandanten-Tabelle |
| ErfassungsperiodID | GUID |  | | Erfassungsperiode |
| Km | Double | |  | Längswert des Bezugspunktes |
| SectorLength | Double | |  | Länge |
| Name | String | |  | Textueller Name |
| Sequence | String | |  | Sequenz-Nummer |
| Operation | Int | | 2 (Create), 1 (Update) 0 Delete | Operation aus Interlis |
| MarkerGeom | Geometry (Point) | |  | Punkt-Geometrie des Sektors |
| ImpNr | Long | | 1,2,3… | EMSG-seitig vergebene Import-Nummer |

ACHSUPDATECONFLICT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feld** | **Typ** | | **Wertebereich** | **Bedeutung** |
| ID | GUID | |  | Primärschlüssel. |
| AchsenSegmentID | GUID | |  | Referenz Achsensegment |
| MandantID | GUID |  | | Zuordnung zu Mandanten-Tabelle |
| Name | String | |  | Beschriftung |
| Sequence | String | |  | Sequenz-Nummer |
| ConflictType | Int (Bitmuster) | | 2: Neues Achssegment,  4: Problem bei Übernahme Strassen-Abschnitt,  8: Problem bei Übernahme Zustand | Konflikt-Typ |

#### ****Caching****

Es findet kein serverseitiges Caching statt – weder von externen WMS Diensten, noch von dynamischen Bearbeitungslayern am EMSG-Master. Clientseitig werden die Abfragen in Tiles durchgeführt, was ein temporäres clientseitiges Caching erlaubt.

#### Achssegmentreferenzen – Struktur

Das Datenmodell in Abbildung 42 illustriert den Zusammenhang zwischen Fachdaten und deren Referenzen (Netzbezug) auf die Achsen, bzw. Achssegmente. Es ist aus Gründen der Übersichtlichkeit attributiv nicht vollständig und hat das Ziel die strukturellen Zusammenhänge der GIS-Entitäten zu veranschaulichen.



Abbildung 42: EMSG-Master GIS-Datenmodell Struktur

Das Datenmodell für den GIS-Modus des EMSG-Masters besteht im Wesentlichen aus drei logischen Ebenen.

* **EMSG – Achsen**  
  Diese werden aus dem Basissystem über die INTERLIS 2 Schnittstelle bezogen und werden durch die EMSG Anwendung in Folge nicht mehr manipuliert (mit Ausnahme bei der Aktualisierung der Achsen – siehe Abschnitt 4.3.2). Für die Anwendungsfälle im EMSG-Master besteht keine Notwendigkeit die Entitäten AchseGIS und SektorGIS im DBMS vorzuhalten – es wird ausschliesslich die Geometrie der AchsSegmenteGIS benötigt (siehe auch Abschnitt 4.4.7.1)
* **EMSG Referenzen**  
  Definiert die Referenzen auf die Achssegmente. Die Referenzen werden durch die EMSG Anwendung erzeugt und bei den entsprechenden Bearbeitungsvorgängen nachgezogen. Mit Hilfe der Entität AchsenReferenzGIS und ReferenzGruppGIS wird eine n:m Assoziation zwischen EMSG Achsen und EMSG Fachdaten ermöglicht. Die Entität AchsenReferenzGIS definiert den räumlichen Bezug auf ein einzelnes AchsSegment (AchsSegmentGIS). Die Tabelle ReferenzGruppGIS erlaubt es, mehrere AchsenReferenzGIS zu gruppieren, bzw. zusammenzufassen.
* **EMSG Fachdaten**  
  Beinhaltet die für EMSG Anwendung wesentlichen Fachdaten. Diese beziehen sich über die EMSG Referenzen auf die Achssegmente. Alle Fachdaten haben eine eigene Geometrie (MULTILINESTRING), die sich aus den Achssegmentgeometrien ableitet. Die eigene Geometrie der EMSG Fachdaten ist prinzipiell redundant zu den EMSG Referenzen, ist aber aus Performance- und Verarbeitungsgründen notwendig und sinnvoll.

Im Wesentlichen stellen die EMSG Referenzen einen 1:n Bezug zwischen EMSG Fachdaten und EMSG Achsen her. Geometrische Abhängigkeiten zwischen EMSG Fachdaten werden nicht in der Datenbank abgebildet, sondern bei Eingabe durch den Nutzer und anschliessender Prüfung durch die EMSG Geschäftslogik.

### Inspektionsrouten und Check-Out von Daten

Inspektionsrouten werden am EMSG-Master definiert, um eine Menge an Strassenabschnitten zusammenzufassen um diese schlussendlich auf EMSG-Mobile für eine weitere Bearbeitung exportieren zu können (Check-Out).

Inspektionsrouten können beliebig viele Strassenabschnitte enthalten, verweisen aber immer auf einen vollständigen Strassenabschnitt und nicht auf Teile von diesem. Über die Zwischentabelle InspektionsrouteStrassenabschnitteGIS wird eine n:m Relation zwischen Strassenabschnitten und Inspektionsrouten definiert. Dass ein aktueller Strassenabschnitt nur einer Inspektionsroute zugeordnet werden kann wird durch die EMSG Geschäftslogik sichergestellt[[27]](#footnote-28).

Abbildung 43 stellt das entsprechende Datenmodell dar:

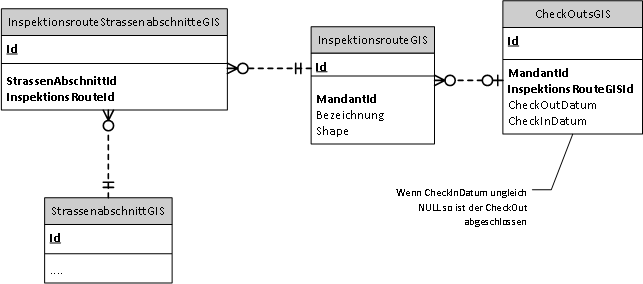


Abbildung 43: Datenmodell Inspektionsrouten und Check-Outs

### Datenmodell EMSG-Mobile

Für die Bearbeitung im EMSG-Mobile müssen folgende Typen von Daten vom EMSG-Master zur Verfügung gestellt werden:

* Rasterdaten (Hintergrundbilder)
* Vektordaten/Fachdaten

Für die Abbildung von Vektordaten wird ein identes Subset des EMSG-Masters Datenmodell verwendet. GIS-Daten und Fachdaten werden in Form von serialisierten Binär-Dateien gespeichert.

#### Fachdaten

Für EMSG-Mobile sind folgende EMSG GIS-Entitäten notwendig:

* Achsen, Achssegmente, Sektoren (für Herstellung von linearen Referenzen und der Definition/Ableitung der Geometrien der EMSG GIS-Entitäten, siehe Abschnitt 4.4.7.1)
* Strassenabschnitte
* Zustandsabschnitte inkl. Schadensformulare, Massnahmenvorschläge

Abbildung 44 stellt das entsprechende Datenmodell dar (Attribute sind aus Gründen der Übersichtlichkeit nicht vollständig abgebildet):



Abbildung 44: Datenmodell EMSG-Mobile GIS Daten (Achsen – Referenzen – Fachdaten)

Die Attribute und deren Ausprägung sind ident zu dem im EMSG-Master Datenmodell (siehe Abschnitte 4.4.4 bis 4.4.8)

#### Rasterdaten

Zusätzlich zu den Vektordaten müssen die entsprechenden Hintergrund-Rasterdaten ebenfalls am EMSG-Mobile vorgehalten werden. Diese Rasterdaten werden aus den WMS-Diensten von swisstopo generiert und liegen in Form von lokalen Karten-Caches vor.

Am EMSG-Mobile stehen folgende Layer zu Verfügung:

* Karte (SW)
* AV-Daten
* Hausnummern

Für die Generierung der Rasterdaten am EMSG-Server wird das Tool BruTile eingesetzt, das aus WMS- und WMTS-Diensten entsprechende lokale Karten-Caches erzeugen kann.

### Datentypen

Falls nicht explizit in den Anforderungen definiert werden folgende Feldlängen bzw. Maximalwerte in der Datenbank verwendet:

* string (einzeilige Eingabefelder) 150 Zeichen
* string (mehrzeilige Eingabefelder) 8‘000 Zeichen
* int 2‘147‘483‘647
* decimal 2‘147‘483‘647.00

## Deployment-Sicht

### Skalierbarkeit

Für den Applikationsserver ist eine Skalierbarkeit gegeben. EMSG-Master kann auf mehreren Instanzen installiert werden. Da einige Daten (siehe 4.6.1.2.1.1) eines Benutzers in der Session (mit Konfiguration In-Process) gehalten werden, muss im Loadbalancing gewährleistet werden, dass der Benutzer über die gesamte Dauer seiner Sitzung der gleiche Applikationsserver zugewiesen wird. In Bezug auf die Mehrsprachigkeit (siehe 4.8.7.2) müssen Änderungen auf jeden Applikationsserver durchgeführt werden da die Ressourcedateien Bestandteil der Applikationslogik sind und nicht zentralisierbar sind.

Das Windows Service (Siehe Kapitel 4.3.2.4), welches die „Kopie Achsdaten“ regelmässig mit Hilfe der INTERLIS 2 Schnittstelle aktualisiert, ist auf Grund des vorgesehenen Prozesses nicht skalierbar.

### Einsatz und Verteilung

Das Deployment der Software erfolgt im ersten Schritt durch die IT auf der Serverlandschaft. Bestandteil der Software ist eine Web-Applikation und ein Windows-Service für den Applikationsserver, eine Servicedefinition für den GeoServer WMS Dienst und dem Schema für die Datenbank. In der Web-Applikation sind dabei bereits die Software für die Rich Client-Anwendung sowie die Sprachübersetzungen integriert.

Das Deployment der Rich Client-Anwendung erfolgt durch den Benutzer selbst auf dem Benutzer-PC. Dazu lädt er von der Web-Applikation das Installationsprogramm für die Rich Client-Anwendung herunter. Das Installationsprogramm muss dann auf den Benutzer-PC z.B. mittels USB Datenträger transferiert werden (sofern dies nicht schon direkt auf den Benutzer-PC heruntergeladen wurde) und dort ausgeführt werden. Das Installationsprogramm übernimmt das Deployment auf dem Benutzer-PC.

### Parametrisierung

Unter anderem können folgende Parameter bei EMSG-Master in der Konfigurationsdatei (web.config) eingestellt werden:

* **Allgemeine Konfiguration**
  1. Datenbank Verbindung
  2. WMS-Verbindung
  3. Etc.
* **Konfiguration IIS**
  1. Session-Timeout
  2. Konfiguration GIS Layer
  3. Zusatzinformationen und Basislayer
  4. GIS-Scales
  5. WMSCacheFolderPath
  6. Etc.

Eine vollständige Übersicht und weitere Details zur Parametrisierung können dem EMSG Konfigurationshandbuch web.config [11] entnommen werden.

### Netzwerkzonen

Dieser Bereich beschreibt die verschiedenen Netzwerkzonen von EMSG. Das Netzwerk-Diagramm in Abbildung 45 beinhaltet eine graphische Darstellung der Netzwerkzonen, welche dann im Anschluss genauer beschrieben werden.

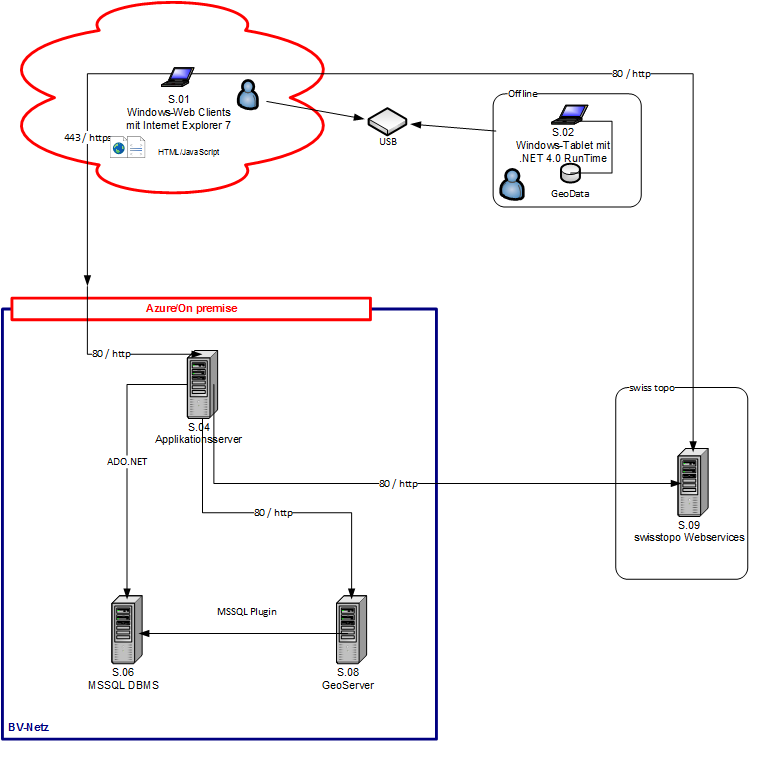


Abbildung 45: Architekturskizze

Der Applikationsserver selbst kommuniziert mit dem MSSQL DBMS.

EMSG-Mobile (rechts oben) verfügt über keine Internet-Verbindung. Sämtliche benötigte Daten werden importiert (z.B. mit einem USB-Stick) und dann Offline gehalten.

## Realisierungs-Sicht

### Schichten des Systems EMSG-Master

Abbildung 46 zeigt die Einteilung von EMSG-Master in seine 3 Schichten: Datenschicht, Geschäftslogik und die Präsentationsschicht.

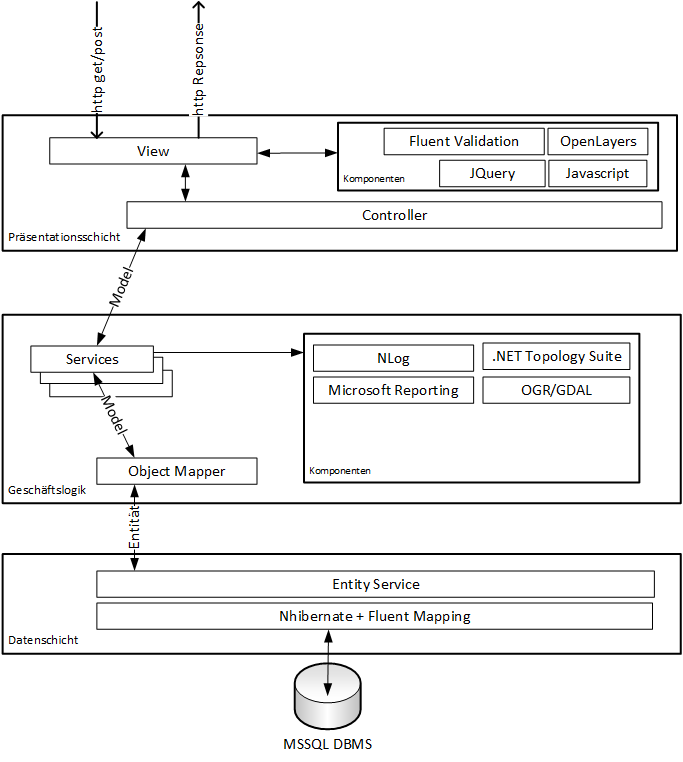


Abbildung 46: EMSG-Master Schichten

Zwischen den einzelnen Schichten (Benutzeroberfläche, Geschäftslogik und Datenschicht) werden Interfaces als Kommunikationsschnittelle verwendet. Auch innerhalb einer Schicht werden Interfaces verwendet um eine hohe Unabhängigkeit der Komponenten und Schichten zu gewährleisten.

#### Datenschicht

Die Datenschicht kümmert sich um das Persistieren der Daten.

##### NHibernate

Alle Zugriffe auf die Applikationsdatenbank erfolgen mittels NHibernate.

Um eine einfache Wartbarkeit für die Datenbankzugriffsoperationen und dem Mapping auf Tabellen zu gewährleisten wird FluentMapping von NHibernate eingesetzt. Das Mapping erfolgt somit rein im Source Code und nicht in externen Dateien wie z.B. XML.

**Alternativer Zugriff via NHibernate Query Language (HQL)**

Für die weiter unten aufgeführten Module wird ein direkter Zugriff durch NHibernate in die Datenbank realisiert um die notwendige Performance gewährleisten zu können. Durch die direkte Verwendung können Stapelaufgaben (Deletes, Copy) direkt in der Datenbank ausgeführt werden und laufen somit nicht über den ORM Teil im EMSG.Master.

HQL wird von folgenden Modulen verwendet (nicht exklusiv, dh. der normale NHibernate Zugriff via Mapping und Modelobjekten erfolgt dennoch):

1. AchsenImportService
2. AchsenUpdateService
3. Jahresabschluss
4. NetzverwaltungGIS (für die initiale Extentberechnung)

##### Fluent Mapping

Diese Komponente ist eine Erweiterung für NHibernate und wird eingesetzt um die Objekte auf Tabellen im Code beschreiben zu können.

##### Entity Service

Der Zugriff auf die Datenbank erfolgt über Entity-Klassen. Diese kapseln den darunterliegenden ORM (NHibernate) und schaffen eine Zentralisierung bzw. einfach Wiederverwendung von Datenbankzugriffsoperationen. Für jede logische Entität gibt es eine eigene Entity-Klasse. Sie beinhalten alle CRUD sowie Filteroperationen.

#### Geschäftslogik

Die Geschäftslogik stellt eine Menge von Services zur Verfügung. Ein Service entspricht dabei einer Klasse in C#. Jedes Service stellt für die Benutzeroberfläche einen Einstiegspunkt dar. Ein Service stellt eine Reihe von Operationen/Methoden für einen logischen Bereich zur Verfügung. Ihre Aufgabe ist es die Anforderungen (Anwendungsfälle) mit ihren Akzeptanzkriterien zu realisieren.

Jede Operation/Methode eines Services kümmert sich dabei in Bezug ihres Aufbaus immer um folgende Aktivitäten:

1. Transformation Model (Eingangsparameter) in eine Entität (mittels Object Mapper)
2. Implementierung der funktionalen Anforderungen. (z.B. nur ein Jahresabschluss pro Mandant pro Jahr)
3. Transformation Entität (Ergebnisse werden mittels Rückgabewert an den aufrufenden Controller zurückgegeben) in ein Model (mittels Object Mapper)

In der Benutzeroberfläche werden keine Entitäten verwendet. Oft können Entitäten nicht 1:1 für eine Webseite verwendet werden, da die Darstellung oft von der Abbildung in der Datenbank abweicht. Deshalb werden der Benutzeroberfläche nur Models zur Verfügung gestellt. Bei den Models handelt es sich dabei um Objekte, die alle Eigenschaften/Daten für die Webseite besitzen. Das kann eine 1:1 Darstellung einer Entität sein, kann aber auch eine Aggregation oder Kompensation von Relationen sein.

##### Services

Hierbei handelt es sich um C# Klassen in denen die eigentliche Geschäftslogik (Anwendungsfälle) implementiert wird. Beispiele dafür sind z.B. der Jahresabschluss oder Import von Strassenabschnitten. Diese Klassen stellen den zentralen Kern der Geschäftslogik dar.

Im Folgenden werden wichtige Services genauer beschrieben.

###### SecurityService

Diese Klasse kümmert sich darum, ob ein Benutzer für einen Mandanten eine Aktion serverseitig ausführen darf. Mit ihr wird auch das Menü gesteuert. Es sind für einen Mandanten nur jene Menüpunkte sichtbar, welche er auf Grund seiner zugeordneten Rollen benötigt.

###### LocalizationService

Diese Klasse kümmert sich darum, die Übersetzungen zur Verfügung zu stellen. Es wird dabei eine auf Microsoft Ressource-Dateien basierende Lösung verwendet. Diese Klasse wird in der Geschäftslogik von anderen Service-Klassen (auch von der Präsentationsschicht für die Benutzeroberfläche) genutzt.

###### ReportingService

Die Aufgabe dieser Klasse ist es die Fachdaten für einen konkreten Jahresabschluss für die Vorschau und Erzeugung von Reports (siehe 4.6.1.2.2.3) zur Verfügung zu stellen.

##### Komponenten

Im folgenden Teil werden die einzelnen Komponenten (dargestellt in Abbildung 46) und ihre Aufgabenbereich genauer beschrieben:

###### Object Mapper

Seine Aufgabe ist die Transformation zwischen Entitäten und den Models. Die Transformation dabei erfolgt mittels .NET Reflection. Seine Aufgabe ist es, ein Objekt in ein anderes Objekt umzuwandeln.

###### NLog

Diese Komponente wird für das Logging verwendet. Mit ihr werden das Anwendungslog (siehe Abschnitt 4.8.3) und das Audit Log (siehe Abschnitt 4.8.4) realisiert.

###### Microsoft Reporting Services

Microsoft Reporting Services wird für die Realisierung der Listen-Auswertungen in allen drei Modi verwendet.

Auswertungen werden im Format von Excel (Format 97-2003) bzw. PDF (A4 Format) als Datei erzeugt und können vom EMSG-Master heruntergeladen werden. Das Layout und der Aufbau (inklusive Unterstützung der Mehrsprachigkeit) der Auswertungen wird in RDLC (Format von Microsoft Reporting Services) beschrieben. Die Erzeugung einer Auswertung erfolgt mittels der Rendering Engine der Microsoft Reporting Services direkt am Applikationsserver. Für Diagramme wird diese Rendering Engine auch als Vorschau auf der Webseite genutzt. Für Tabellen wird anstelle der Rendering Engine das Grid von Telerik genutzt.

Tabelle 4 gibt einen Überblick über die Vorschau und die Erzeugung einer Auswertung.

|  |  |  |
| --- | --- | --- |
|  | **Tabellen** | **Diagramme** |
| **Vorschau** | Einsatz des Grid von Telerik zur Anzeige. Sortierung und Filterung erfolgt mittels Grid von Telerik. | Verwendung der Microsoft Report Rendering Engine zur Einbettung des Diagramms (Grafik) auf der Webseite. Filterung erfolgt ausserhalb des Diagramms auf der Webseite. |
| **Erzeugung von Excel, PDF** | Verwendung der Microsoft Report Rendering Engine zur Erzeugung der Auswertung gemäss den eingestellten Filter- und Sortierkriterien des Grids von Telerik. | Verwendung der Microsoft Report Rendering Engine zur Erzeugung der Auswertung gemäss den eingestellten Filterkriterien der Webseite. |

Tabelle 4: Vorschau und Erzeugung von Auswertungen mittels Microsoft Reporting

###### Reporting GIS

Das Erstellen von Kartenauswertungen im PDF-Format wird wie in Punkt 4.6.1.2.2.3 beschrieben durchgeführt. Die Karten selbst werden als Single-Tile vom GeoServer beziehungsweise von Swisstopo entsprechend der aktuell gewählten Zoomstufe und Extent sowie der Browsergrösse und den sichtbaren Layern exportiert. Dieses Kartenbild wird für die Erstellung der Auswertungen wie ebenfalls im Punkt 4.6.1.2.2.3 beschrieben herangezogen.

Kartenansichten können nur im PDF Format zur Verfügung gestellt werden.

Aufgrund eines Fehlers in den Microsoft Reporting Services können Bilder nur mit sehr viel höherer Auflösung gedruckt werden wenn diese von einer sehr hohen Auflösung herunterskaliert werden. Da diese Bilder ein viel größeres Datenvolumen haben erfolgt die Übergabe an die Reporting Services über eine Temporärdatei. Diese Temporärdatei wird via .net Framework Funktionen erzeugt und auch gelöscht. Die Anlage der Datei erfolgt in der Regel im Temp Verzeichnis des Ausführenden Benutzers (also des Benutzers, der den EMSG.Master Dienst ausführt). Nach Verwendung wird die Temporärdatei gelöscht.

###### .NET Topology Suite

Die .NET Topology Suite ist eine Open Source C# Bibliothek, welche umfangreiche GIS-Funktionalität für .NET zur Verfügung gestellt. Diese wird im EMSG-Master für die Validierung und Generierung der Geometrien, sowie im Zusammenspiel mit NHibernate für die Persistierung der Geometrien eingesetzt.

###### OGR/GDAL

Die OGR/GDAL-Library stellt umfangreiche Funktionen für die Konvertierung von Geodaten in unterschiedliche Formate zur Verfügung. Diese unterstützt auch leistungsstarke Komprimierungsfunktionen für Rasterdaten und Vektordaten. In der Fachapplikation EMSG wird diese Bibliothek für die Generierung von Rasterdaten-Dateien aus WMS Services und für die Generierung von Shapefiles verwendet. OGR/GDAL erlaubt es auch in der Zukunft unterschiedlichste Quelldatenformate (wie. z.B. WMTS) zu unterstützen und bietet damit eine grosse Flexibilität

#### Präsentationsschicht

In dieser Schicht werden die Masken mit ihren Client- und Serververhalten realisiert. Als realisierendes Werkzeug dafür wird das MVC3 (siehe dazu mehr in 4.6.6.2) Framework von Microsoft eingesetzt.

Der Datenaustausch zwischen den Browser und dem Applikationsserver erfolgt auf zwei verschiedene Arten:

* HTML/XHTML
* JSON

In den Masken, die Formulare darstellen, erfolgt der Datenaustausch über HTML/XHTML. Hingegen wird in den Masken, in denen die Karte dargestellt wird, mittels JSON zwischen dem Browser und dem Applikationsserver kommuniziert.

Am Browser selbst erfolgt die Kommunikation mit der Skriptsprache JavaScript bzw. über die Klassenbibliothek JQuery.

Bei jeder Kommunikation zwischen Browser und Applikationsserver wird serverseitig eine Securityprüfung durchgeführt, ob der Benutzer für den gewählten Mandanten die Aktion ausführen darf (hierbei wird das SecurityService von der Geschäftslogik genutzt – siehe Abschnitt 4.6.1.2.1.1). Die Integration der Security-Prüfung erfolgt in MVC3 am Controller mittels Filter. Zusätzlich wird serverseitig eine Transaktion zur Datenbank aufgebaut. Wenn die Aktion am Applikationsserver erfolgreich durchgeführt werden konnte, findet ein Commit auf die Datenbank statt. Anschliessend werden die Daten in HTML/XHTML bzw. JSON zum Browser zurückgeschickt. Die Integration der Transaktion erfolgt wie die Security-Prüfung mittels Filter in MVC.

**Anmerkung zu Filter in MVC3:**

Hierbei handelt es sich um benutzerdefinierte Klassen die eine programmgesteuerte Möglichkeit zum Einfügen von Anweisungen vor und nach dem Aufruf einer Aktion bieten. Im Falle von Securityprüfungen werden die Anweisungen (=Implementierung der konkreten Securityprüfung) davor ausgeführt (nur wenn die Securityprüfung erfolgreich ist soll auch die Aktion ausgeführt werden). Hingegen bei der Transaktion auf die Datenbank werden die Anweisungen (=Implementierung von Commit einer Transaktion) nach der Aktion ausgeführt.

Der Vorteil von Filtern liegt in der einfachen Wiederverwendung in unterschiedlichen Aktionen und Controllern.

#### Controller

Klassen, die Aktionen für den Client zur Verfügung stellen.

#### View

Beschreibung des Aufbaus und Inhalts der Webseiten in XML bzw. Razor Syntax.

#### Komponenten

In den folgenden Unterabschnitten werden die einzelnen Komponenten (dargestellt in Abbildung 46) und ihre Aufgabenbereich genauer beschrieben.

##### Fluent Validation

Mit dieser Komponente wird die Validierung der Models mit Unterstützung der Mehrsprachigkeit durchgeführt. Im Model wird zentral die Validierungslogik deklariert. Die Komponente erzeugt den notwendigen JavaScript-Code für die Validierung am Client sowie serverseitigen Code zur Validierung am Applikationsserver.

##### OpenLayers

OpenLayers ist eine Open Source JavaScript-Bibliothek, die umfangreiche Funktionen für die Darstellung und die Manipulation von GIS-Daten zur Verfügung stellt. Es werden unter anderem auch WebMapServices und WebFeaturesServices nach dem OGC-Standard unterstützt. Ausserdem erlaubt es eine einfache Darstellung von Web Map Tiles Services.

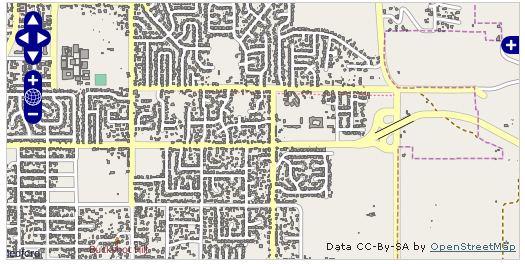


Abbildung 47: OpenLayers JavaScript Kartenelement – Beispiel für Visualisierung von Rasterdaten (WMTS) und Vektordaten (WFS)

##### JQuery

Diese Klassenbibliothek wird eingesetzt um asynchrone Postbacks mittels AJAX zu realisieren.

##### JavaScript

Diese Skriptsprache im Browser wird für den Multimonitorbetrieb (siehe 4.6.6.1) eingesetzt.

### Interaktion der verwendeten Komponenten in EMSG-Master

Die Interaktion zeigt das Zusammenspiel der wichtigsten Komponenten in EMSG-Master. Diese Interaktion spiegelt einen Server-Round-Trip wieder. Gezeigt wird das ganze am Beispiel von Load (siehe Abbildung 48) und Save (siehe Abbildung 49).

Bei jedem Server-Round-Trip wird zuerst der Transaction Scope für NHibernate initialisiert. Es wird immer eine Transaction erzeugt. Eine Connection zu MSSQL wird erst dann geöffnet wenn die ersten Lese- oder Schreibaktivitäten durchgeführt werden.

Im nächsten Schritt wird eine Securityprüfung durchgeführt. Die Securityprüfung bestimmt ob der Benutzer den Aufruf (http Request) durchführen darf.

Anschliessend erzeugt der Model Binder ein typisiertes Objekt welcher der Action des Controllers als Eingangsparameter zur Verfügung gestellt wird. Der Model Binder führt dabei eine strukturelle Validierung der gesendeten Attribute durch.

Im Anschluss wird die Action durch das Framework aufgerufen. Der Action steht das typisierte Objekt zur Verfügung. In einer Action befindet sich die Implementierung eines Anwendungsfall bzw. ruft und verwendet Klassen welche die Implementierung wiederspiegeln.

Am Ende wird die Transaction entweder durch ein Commit oder Rollback beendet. Zweiteres tritt ein falls es zu einem unerwarteten Fehler gekommen ist (z.B. Programmierfehler, Timeout, etc.).

#### Load Verhalten



Abbildung 48: Load Verhalten

#### Save Verhalten



Abbildung 49: Save Verhalten

#### Proxy

EMSG-Master beinhaltet einen Proxy für die Kommunikation Client 🡪 GeoServer bzw. swisstopo und AV. Die Kommunikation der Clients erfolgt somit zentral über den EMSG-Master Server.

Die Clients schicken somit die einzelnen Tile/Map Requests an den EMSG-Master Server, die Zuordnung zu den einzelnen Teilbereichen übernimmt der WMSController. Dieser Controller ersetzt seine eigene URL durch die entsprechenden WMS URLs der Server (URL Rewriting) und führt die Abfrage in Vertretung des Clients durch. Alle Ergebnisse werden transparent dem Client zur Verfügung gestellt.

Für den Client ist nicht ersichtlich, dass die Abfrage an einen anderen Server weitergeleitet wird.



Abbildung 50: EMSG Proxyfunktion

### Schichten des Systems EMSG-Mobile

EMSG-Mobile wird wie EMSG-Master in einer 3-Schichten-Architektur (Datenschicht, Geschäftslogik und Präsentationsschicht) realisiert.

Im Gegensatz zu EMSG-Master wird im EMSG-Mobile keine Datenbank für die Persistierung verwendet. Stattdessen kommen DTOs (DataTransferObjects, siehe Punkt 4.6.3.3) zum Einsatz. Diese DTOs stellen erweiterte EMSG-Master Models dar, die serialisiert im Exportpaket enthalten sind. Die Daten sind somit direkt ohne Konvertierung zur Verfügung und können auf dem gleichen Weg zurück importiert werden. Die Verwaltung der DTOs übernimmt die Datenschicht.

#### Präsentationsschicht

In dieser Schicht werden die Oberflächen mit ihrem Verhalten realisiert. Als zu grunde liegendes Designpattern kommt hier MVVM (Model, View and ViewModel) zum Einsatz. Dies unterscheidet sich vom EMSG-Master da die grundlegende Technologie (WPF und WinForms für „klassische“ Anwendungen) besser behandelt werden kann.

Die Benutzeroberfläche (View) wird in diesem Fall von den ViewModels befüllt, die die Funktion des Controllers aus MVC übernehmen und erweitern bzw. vereinfachen. Die zugehörigen Datenobjekte (Models) sind separat definiert und basieren auf DTOs (DataTransferObjects, siehe 4.6.3.3).

#### Geschäftslogikschicht

Die Geschäftslogik stellt eine Menge von Services zur Verfügung. Ein Service entspricht dabei einer Klasse in C#. Jedes Service stellt für die Benutzeroberfläche einen Einstiegspunkt dar. Ein Service stellt eine Reihe von Operationen/Methoden für einen logischen Bereich zur Verfügung. Ihre Aufgabe ist es die Anforderungen (Anwendungsfälle) mit ihren Akzeptanzkriterien zu realisieren.

Jede Operation/Methode eines Services kümmert sich dabei in Bezug ihres Aufbaus immer um folgende Aktivitäten:

1. Übergabe bzw. Transformation von DTOs in Entitäten für die Benutzeroberfläche
2. Implementierung der funktionalen Anforderungen. (z.B. Speichern des Änderungsprotokolls, Import eines EMSG-Master Exports)
3. Synchronisierung Karte und Daten

Im Folgenden werden wichtige Services genauer beschrieben.

##### DTOService

Diese Klasse ist für die Manipulation der DTO zuständig, also für Laden und Speichern von Exportpaketen. Hier erfolgt die Serialisierung und Deserialisierung der Daten, Nummerierung und Zugriff auf einzelne Objekte.

Die Update/Delete Protokollierung findet ebenfalls in dieser Klasse statt.

##### ExportService

Diese Klasse ist für die Konvertierung der aktuell geöffneten Daten in ein Rückimportsfile zuständig. Dabei wird über einen Filter jedes Element serialisiert und gepackt, in dem eine Änderung erfolgt ist. Hierbei wird nur der letzte Zustand des Objekts übertragen, nicht aber die Reihenfolge oder die vorhergegangenen Zustände.

##### ImportService

Diese Klasse ist für die Verifizierung und Vorbereitung der Daten aus einem EMSG-Master Exportpaket zuständig.

Hier erfolgt eine Basisverifizierung (Struktur, Fehlerprüfung auf Transferfehler).

Ist das Paket in Ordnung kann die bestehende Dateistruktur (sofern vorhanden) verworfen und die Dateien und Daten aus dem aktuellen Paket entpackt werden. Diese Daten stehen dem EMSG-Mobile von diesem Zeitpunkt an zur Verfügung.

#### Datenschicht

Wie schon eingangs beschrieben ist die Datenschicht für die Bereitstellung der Daten, die Konvertierung von DTO in Models und Persistierung als DTO zuständig.

Ein DataTransferObject (DTO) ist eine erweiterte Objektklasse aus dem EMSG-Master. Die Entitäten, die in der EMSG-Master Datenbank persistiert werden können in ein Transferobjekt umgewandelt, serialisiert und in einem Exportpaket gespeichert werden. Die in der Datenschicht deserialisierten Objekte stehen 1:1 wie am EMSG-Master zur Verfügung und können über die gleiche Logik wieder importiert werden. Dadurch ist sichergestellt, dass keine Konvertierungs- oder Transformationsfehler auftreten können.

Die für das Kartencontrol Vektordaten sind im DTO enthalten und werden dem Kartencontrol im GeoJson Format übergeben. Die Rasterdaten im lokalen Karten-Cachen werden wie ein WMTS behandelt und vom Kartencontrol dargestellt.

### Interaktion der verwendeten Komponenten in EMSG-Mobile

Die Interaktion zeigt das Zusammenspiel der wichtigsten Komponenten in EMSG-Mobile. Gezeigt wird das ganze am Beispiel von Import (Load, siehe Abbildung 48) und Export (Save, siehe Abbildung 49).

Bei jeder Aktion wird das entsprechende Service initialisiert, für Import das ImportService, für Export das ExportService (im folgenden einfach Service genannt). Da EMSG-Mobile über keine Datenbankverbindung verfügt wird als Datenverbindung dieser Service herangezogen.

Beide Dienste sind für das Lesen bzw. Erstellen der entsprechenden Dateien vorgesehen und implementieren auch das eigentliche Dateiformat.

#### Load Verhalten

Alle zum EMSG-Mobile versandten Daten werden in einer Transferdatei zusammengefasst und vom ImportService in ein temporäres Verzeichnis entpackt. Der Inhalt der Datei wird mit dem PackageDescriptor verglichen und die Datei damit verifiziert. Danach beginnt das Bereitstellen der einzelnen Objekte. Sobald die Daten komplett bereitgestellt wurden wird die Karte initialisiert und die DTOs (DataTransferObjects) geladen. Unmittelbar danach ist EMSG-Mobile einsatzbereit.



Abbildung 51: Ladeverhalten in EMSG.Mobile

#### Save Verhalten

Alle Daten, die der Benutzer geändert hat wurden während der Arbeit mit EMSG-Mobile als geändert markiert. Der ExportService iteriert über alle Objekte, die im Speicher vorhanden sind und extrahiert eine Liste aller geänderten Daten. Diese Datenliste wird zusammen mit allen anderen notwendigen Dateien in Protokollform zusammen gepackt und als Exportdatei zur Verfügung gestellt.



Abbildung 52: Speicherverhalten

### Code-Sicht

#### Namensgebung

Die Namensgebung für Architektur-Komponenten basiert auf der englischen Sprache. Für fachliche Ausdrucke wird jedoch Deutsch verwendet.

Beispiele: „StrassenabschnittMapping“, „IRealisierteMassnahmeGISModelService“

#### Dependency Injection

Hierbei handelt es sich um ein Entwurfsmuster (Pattern), um Abhängigkeiten in objektorientierten Systemen zwischen den einzelnen Komponenten so gering wie möglich zu halten. Die einzelnen Komponenten kommunizieren dabei über vordefinierte Schnittstellen miteinander und sind losgelöst von konkreten Implementierungen.

Dependency Injection wird in EMSG mittels Autofac umgesetzt. Autofac bietet eine einfache Möglichkeit, um Implementierungen von Schnittstellen zu registrieren und zur Laufzeit Instanziierungen vorzunehmen.

### Benutzeroberfläche EMSG-Master

Dieser Bereich beschreibt die technische Realisierung der Benutzeroberfläche. Eine Übersicht über die verschiedenen Masken und der Navigation befindet sich im Usability Konzept [9] bzw. in den Systemanforderungen [7].

#### Multimonitorbetrieb

Das Ziel des Multimonitorbetrieb ist es den Teile des Inhalt einer Webseite auf ein weiteres eigenständiges Fenster auszulagern. Die Anforderung bestand darin diese Funktionalität für die Auswertungen im GIS Modus anzubieten.

Die Verwendung des Multimonitorbetrieb ist optional. D.h. in den vorgesehenen Anwendungsfällen besteht wie bei anderen Anwendungsfällen die Möglichkeit den ganzen Inhalt einer Webseite wie gewohnt in einem Fenster darzustellen.

Der Multimonitorbetrieb ist so gestaltet dass in einem weiteren Fenster Filterkriterien und die Tabellenansicht ausgelagert werden. Diese Fenster besitzt kein Menü. Die Navigation in der Applikation erfolgt rein über das Hauptfenster.

Die Realisierung des Multimonitorbetrieb erfolgt in JavaScript unter der Verwendung von JQuery am Client. Dies beinhaltet das Öffnen und Schliessen des Fenster sowie die Synchronisation der beiden Fenster. Serverseitig ist keine Infrastruktur notwendig wodurch am Applikationsserver keine weitere Last entsteht.

Die Synchronisation erfolgt immer dann wenn in einem Fenster eine Änderung vorgenommen wird welche eine Aktualisierung des Inhalts zur Folge hat. Die Synchronisation sieht vor dass ein Fenster auf Änderungen des anderen Fensters reagiert und die Filterkriterien aktualisiert. Auf Basis der Filterkriterien ist jedes Fenster selbst dafür verantwortlich den Inhalt am Applikationsserver neu abzufragen und zu aktualisieren. Die Synchronisation am Client wird somit einfach gehalten und erfordert daher wenig Ressourcen.

In Abbildung 53 ist das Konzept des Multimonitorbetriebs inklusive der Synchronisierung dargestellt.



Abbildung 53: Konzept Multimonitor

#### Einsatz von Microsoft MVC3 Framework

Aus folgenden Gründen in Bezug auf die Präsentationsschicht und Geschäftslogik wurde entschieden, für EMSG-Master das MVC3 Framework von Microsoft in der Applikation einzusetzen:

* Komplexität kann in Model, View, Controller aufgeteilt werden
* REST - Anfragen an die Applikation sind „stateless“
* Bessere, automatisierte Testbarkeit
* Trennen der Aufgabebereiche (z.B. UI-Design, Validierung, Geschäftslogik, CSS)



Abbildung 54: MVC3

Diese Aufteilung (siehe dazu auch Abbildung 54) der Aufgaben sieht folgendermassen aus:

* **Model**: Das Model stellt die Daten dar. Dies wird auch oft als Fachdaten (z.B. Strassenabschnitt) bezeichnet. In dem Model ist keine Geschäftslogik abgebildet.
* **View**: Die View kümmert sich um die Anzeige der Daten des Models. Die View bietet im Browser Interaktionsmöglichkeiten wie z.B. die Eingabe oder das Editieren von Daten an. In einer View ist wie im Model selbst keine Geschäftslogik abgebildet.
* **Controller**: Der Controller fungiert als ein Mediator zwischen View und Model. Seine Aufgabe ist es, die Daten des Models an die View weiterzureichen. Er verarbeitet die Benutzeraufgaben und leitet diese an die Geschäftslogik weiter.

Im Folgenden wird das Zusammenspiel beispielhaft erläutert. Im ersten Schritt werden die Daten eines Abschnitts angezeigt. Im zweiten Schritt wird die Belastungskategorie geändert und das System schlägt die Standardbreite vor.



Abbildung 55: Beispiel MVC3

#### Funktionales Testen für den Applikationsserver

Für die funktionalen Tests wird SpecFlow[[28]](#footnote-29) eingesetzt. Als Einstiegspunkt für die Tests werden die Actions der Controller verwendet. Um nahe der realen Umgebung zu testen wird die Library MvcIntegrationTestFramework verwendet. Mit dieser Library werden Applikationsserver und Browser simuliert.



Abbildung 56: Konzept Funktionales Testen

Folgende konkrete Vorteile ergeben sich durch diese Library:

* Unterscheidung zwischen GET- und POST-Request
* Filter werden behandelt bzw. abgearbeitet (z.B. Transaktion, Security, etc.) und brauchen nicht simuliert werden
* Serverseitige Validierung bei der Generierung des Models
* Das Result beinhaltet den ModelViewState, das ActionResult und HTML Code

Die Library wird nur für das Testen verwendet. Für den Einsatz von EMSG ist sie nicht notwendig. Sie ist daher auch nicht Bestandteil des Deployment.

#### Partial Rendering

In EMSG-Master werden Anfragen wenn möglich asynchron durchgeführt. Realisiert wird dies durch die AJAX-Technologie. Durch die asynchronen Anfragen wird partial Rendering ermöglicht. Dadurch ergeben sich folgende Vorteile:

* Der Webserver muss nur die betroffenen Elemente rendern und nicht die ganze Seite, da sich nur ein Teilbereich darin ändert.
* Es werden nur die Daten zum Client geschickt, die sich geändert haben
* Schnellerer Seitenaufbau im Browser
* Kein störendes Flackern im Browser beim Seitenaufbau

#### CSS und Images

Images und Formatierungen des Layouts werden zentral abgelegt, um eine einfache und schnelle Wartbarkeit zu gewährleisten.

#### Berechnungslogik

Es gibt keine client-seitige Berechnungslogik, um eine doppelte Implementierung der Berechnungen auf Client und Server zu vermeiden.

### Benutzeroberfläche EMSG-Mobile

Dieses Bereich beschreibt die technische Realisierung der Benutzeroberfläche. Eine Übersicht über die verschiedenen Masken und der Navigation befindet sich im Usability Konzept [9] bzw. in den Systemanforderungen [7].

#### Einsatz von MVVM

Das Pattern MVVM (Model-View-ViewModel) basiert auf den Prinzip von MVC3 und zielt dabei speziell auf Silverlight und WPF ab. Es eignet sich daher für den Einsatz in EMSG-Mobile.

Wie in MVC3 bietet dieses Pattern eine saubere Trennung der Zuständigkeiten und bietet eine Basis für bessere, automatisierte Testbarkeit.

## Infrastruktur- und Hardware-Sicht

### Verwendete Produkte und Methoden

Bezüglich Angaben zu lizenzpflichtigen Produkten siehe Tabelle 6 und Tabelle 7.

#### Methoden

|  |  |  |
| --- | --- | --- |
| **Methode** | **Herkunft** | **Beschreibung / Zweck und Einsatzgebiet** |
| Scrum | Scrum Alliance | Empirischer Ansatz zur Entwicklung komplexer Software-Systeme, Details siehe http://www.scrumalliance.org/ |
| HERMES | Bund | Schweizerische Projektführungsmethode, Details siehe http://www.hermes.admin.ch/ |
| Continuous Integration | Kent Beck et al. | Siehe http://de.wikipedia.org/wiki/Kontinuierliche\_Integration |

Tabelle 5: Methoden

#### Fertigkomponenten Entwicklung

Für EMSG-Master werden folgende Fertigkomponenten eingesetzt:

| **Produkt** | **Hersteller und Lizenzinfo** | **Kosten-pflichtig** | **Beschreibung / Zweck und Einsatzgebiet** |
| --- | --- | --- | --- |
| .NET 4.5 Framework | Microsoft | Nein | Entwicklungs-Framework |
| MVC3 mit Razor Generator | Microsoft | Nein | Details siehe http://www.asp.net/mvc |
| Reporting Services | Microsoft | Nein | Framework für Auswertungen |
| NHibernate | OSS (GNU) | Nein | OR-Mapping-Komponente, Details siehe <http://nhforge.org> |
| Fluent Mapping | OSS (BSD) | Nein | Ermöglicht das vollständige Mapping von Entitäten in NHibernate direkt im Code anstelle von Code und XML  Details siehe <http://fluentnhibernate.org> |
| Autofac | OSS (MIT) | Nein | Komponente zum Handling der Dependency Injection, Details siehe http://code.google.com/p/autofac |
| JQuery | OSS (MIT/GPL) | Nein | JavaScript Bibliothek (für AJAX, etc.), Details siehe <http://jquery.com> |
| OpenLayers 2.11 | OSS (FreeBSD) | Nein | OpenLayers ist eine leistungsstarke und einfach erweiterbare JavaScript Bibliothek für Darstellung und Bearbeitung von GeoDaten in einem Webclient. In der Fachapplikation EMSG wird die Bibliothek für die Darstellung der Geodaten im EMSG-Master verwendet.  Details siehe www.openlayers.org |
| GeoServer 2.9.1 | OSS (GPL) | Nein | Produkt für die Verwaltung und Darstellung raumbezogenen Informationen. |
| .NETTopologySuite | OSS (LGPL) | Nein | Die .NETTopologySuite ist ein C# Port der sehr verbreiteten Java Topology Suite (JTS). Diese Bibliothek stellt GIS Funktionalität für C#/.NET zur Verfügung und erlaubt eine einfache Persistierung über die Komponente NHibernate. |
| BruTile | OSS (Apache 2.0) | Nein | BruTile ist eine Bibliothek die dazu dient auf gekachelte Karten-Dienste zuzugreifen. Für EMSG wurde diese Bibliothek erweitert um u.a. eine bessere Unterstützung von WMS zu erreichen. BruTile wird verwendet um für EMSG-Mobile einen lokal verwendbaren offline Karten-Cache zu erstellen.  Details siehe https://brutile.codeplex.com/ |
| NLog | OSS (BSD) | Nein | Komponente zum Loggen von Informationen in Dateien bzw. Datenbanken  Details siehe http://nlog-project.org |
| Fluent Validation | OSS (Apache 2.0) | Nein | Details siehe http://fluentvalidation.codeplex.com |
| MVCIntegrationTestFramework | TechTalk | Nein | <https://github.com/techtalk/MvcIntegrationTestFramework> |
| Telerik | Telerik[[29]](#footnote-30) | Ja | Controls für MVC (z.B. Grid, FileUpload, etc.)  Details siehe http://www.telerik.com |
| SpecFlow | OSS (BSD) | Nein | Kommt für Testen zum Einsatz. Anforderungen werden in Gherkin Format beschrieben. Mapping der Anforderungen auf die Geschäftslogik durch Code (Binding). Ausführen der Tests mittels NUnit Testframework.  Details siehe <http://www.specflow.org> |
| Moq | OSS (BSD) | Nein | Wird für Unit Tests verwendet um Platzhalter/Attrappen für echte Objekte zu erstellen.  Details siehe <http://code.google.com/p/moq/> |
| NUnit | OSS (zlib/libpng) | Nein | Testframework für .NET Programmiersprachen  Details siehe <http://www.nunit.org> |
| MapProxy | OSS (Apache 2.0) | Nein | Wird verwendet um WMTS Services für Reports zur Verfügeung zu stellen, in dem sie als WMS ausgegeben werden (Karten-Reports können momentan nur WMS verwenden) |
| Apache Tomcat | OSS (Apache 2.0) | Nein | Open-Source-Webserver und Webcontainer, der verwendet wird um GeoServer zu hosten |
| Apache HTTP Server | OSS (Apache 2.0) | Nein | Webserver der verwendet wird um Mapproxy zu hosten |

Tabelle 6: Fertigkomponenten Entwicklung EMSG-Master

Für EMSG-Mobile werden folgende Fertigkomponenten eingesetzt:

|  |  |  |  |
| --- | --- | --- | --- |
| **Produkt** | **Hersteller oder OSS** | **Kosten-pflichtig** | **Beschreibung / Zweck und Einsatzgebiet** |
| .NET 4.0 Framework | Microsoft | Nein | Entwicklungs-Framework |
| WPF (Windows Presentation Foundation) | Microsoft | Nein | Framework für Windows Smart Client-Benutzeroberflächen |
| NLog | OSS (BSD) | Nein | Komponente zum Loggen von Informationen in Dateien bzw. Datenbanken  Details siehe http://nlog-project.org |
| OpenLayers 2.11 | OSS (FreeBSD) | Nein | OpenLayers ist eine leistungsstarke und einfach erweiterbare JavaScript Bibliothek für Darstellung und Bearbeitung von GeoDaten in einem Webclient. In der Fachapplikation EMSG wird die Bibliothek für die Darstellung der Geodaten im EMSG-Mobile verwendet.  Details siehe www.openlayers.org |

Tabelle 7: Fertigkomponenten Entwicklung EMSG-Mobile

#### Werkzeuge Entwicklung

Als unterstützendes Werkzeug für die Entwicklung kommt Microsoft Team Foundation Server zum Einsatz. Mit diesem Werkzeug werden folgende Bereiche abgedeckt bzw. realisiert:

* Source Control Management
* Work Item Tracking (Verwaltung von Fehlermeldungen)

Zusätzlich werden folgende weiteren Werkzeuge für die Entwicklung verwendet:

* Continuous Integration (TeamCity von Jetbrains)
* ReSharper (von Jetbrains)
* Team Foundation Server Tools für Microsoft Visual Studio
* IIS Express für die Entwicklung des Applikationsserver
* Macrobject Word-2-CHM 2009 Pro (für die Generierung der Online-Hilfe aus dem Anwendungshandbuch)
* ResX Localization Studio 2 (für die Bearbeitung der Übersetzungen in den Ressource-Dateien)

#### Fertigkomponenten Produktion

Für EMSG-Master werden folgende Fertigkomponenten eingesetzt:

| **Produkt** | **Hersteller und Lizenzinfo** | **Kosten-pflichtig** | **Beschreibung / Zweck und Einsatzgebiet** |
| --- | --- | --- | --- |
| .NET 4.5 Framework | Microsoft | Nein | Entwicklungs-Framework |
| MVC3 mit Razor Generator | Microsoft | Nein | Details siehe http://www.asp.net/mvc |
| Reporting Services | Microsoft | Nein | Framework für Auswertungen |
| NHibernate | OSS (GNU) | Nein | OR-Mapping-Komponente, Details siehe <http://nhforge.org> |
| Fluent Mapping | OSS (BSD) | Nein | Ermöglicht das vollständige Mapping von Entitäten in NHibernate direkt im Code anstelle von Code und XML  Details siehe <http://fluentnhibernate.org> |
| Autofac | OSS (MIT) | Nein | Komponente zum Handling der Dependency Injection, Details siehe http://code.google.com/p/autofac |
| JQuery | OSS (MIT/GPL) | Nein | Javascript Bibliothek (für AJAX, etc.), Details siehe <http://jquery.com> |
| OpenLayers 2.11 | OSS (FreeBSD) | Nein | OpenLayers ist eine leistungsstarke und einfach erweiterbare JavaScript Bibliothek für Darstellung und Bearbeitung von GeoDaten in einem Webclient. In der Fachapplikation EMSG wird die Bibliothek für die Darstellung der Geodaten im EMSG-Master verwendet.  Details siehe www.openlayers.org |
| GeoServer | OSS (GPL) | Nein | Produkt für die Verwaltung und Darstellung raumbezogenen Informationen. |
| .NETTopologySuite | OSS (LGPL) | Nein | Die .NETTopologySuite ist ein C# Port der sehr verbreiteten Java Topology Suite (JTS). Diese Bibliothek stellt GIS Funktionalität für C#/.NET zur Verfügung und erlaubt eine einfache Persistierung über die Komponente NHibernate. |
|  |  |  |  |
| BruTile | OSS (Apache 2.0) | Nein | BruTile ist eine Bibliothek die dazu dient auf gekachelte Karten-Dienste zuzugreifen. Für EMSG wurde diese Bibliothek erweitert um u.a. eine bessere Unterstützung von WMS zu erreichen. BruTile wird verwendet um für EMSG-Mobile einen lokal verwendbaren offline Karten-Cache zu erstellen.  Details siehe https://brutile.codeplex.com/ |
| NLog | OSS (BSD) | Nein | Komponente zum Loggen von Informationen in Dateien bzw. Datenbanken  Details siehe http://nlog-project.org |
| Fluent Validation | OSS (Apache 2.0) | Nein | Details siehe http://fluentvalidation.codeplex.com |
| Telerik | Telerik[[30]](#footnote-31) | Nein | Controls für MVC (z.B. Grid, FileUpload, etc.)  Details siehe http://www.telerik.com |

Abbildung 57: Fertigkomponenten Produktion EMSG-Master

Für EMSG-Mobile Es werden die gleichen Komponenten wie in der Entwicklung (siehe Abschnitt 4.7.1.2) verwendet.

## Design-Strategien EMSG-Master

### Locking

Beim Datenzugriff findet kein Locking statt.

### Validierung

Bei der Validierung wird zwischen struktureller und logischer Validierung unterschieden.

#### Strukturelle Validierung

Diese Validierung der Objekte erfolgt bereits in der Präsentationsschicht. Die Validierung erfolgt dabei im Ersten Schritt am Client durch JavaScript. Im zweiten Schritt erfolgt die Validierung am Applikationsserver beim Model Binding. Realisiert wird die Validierung durch das Validierungsframework Fluent Validation (siehe Abschnitt 4.6.1.3).

Folgende strukturelle Validierungen werden auf den Objekten durchgeführt:

* Einhaltung der Wertebereiche (z.B. Bereich, Min, Max, etc.)
* Einhaltung der Katalogwerte
* Pflichtfelder
* Einhaltung des Datentyp

#### Logische Validierung

Neben den strukturellen Validierungen werden auch logische Validierungen durchgeführt. Diese werden nur in der Geschäftslogik durchgeführt. Bei diesen Validierungen werden zur Validierung des Objekts, das gerade angelegt, aktualisiert oder gelöscht werden soll, zusätzlich Daten aus der Applikationsdatenbank oder aus anderen Systemen benötigt. Diese Validierung wird zusätzlich zur strukturellen Validierung in der Geschäftslogik durchgeführt.

### Anwendungslog

Im Anwendungslog (oft auch als technisches Log bezeichnet) werden unerwartete Fehler geloggt. Diese Informationen werden dabei in einer Textdatei am Applikationsserver für eine konfigurierbare Zeitspanne gespeichert (siehe Betriebshandbuch 4.1.3). Pro Logeintrag werden folgenden Informationen gespeichert:

* Name des Applikationsserver
* Windows User Name
* Zeitstempel
* Fehlertext
* Stacktrace
* Fehlerverfolgungsnummer

Falls ein unerwarteter Fehler auftritt, wird am Client eine benutzerfreundliche Fehlerseite dargestellt. Auf dieser Seite wird die Fehlerverfolgungsnummer angezeigt. Diese kann später für die Zuordnung zum Anwendungslog verwendet werden.

Als Komponente zum Loggen wird NLog (siehe Abschnitt 4.6.1.2) eingesetzt.

### Audit Log

Im Audit Log (oft auch als Ereignislog bezeichnet) werden Aktionen wie z.B. das Durchführen der Aktualisierung von Achsen geloggt. Diese Informationen werden in die Applikationsdatenbank gespeichert. Das Einsehen der Informationen erfolgt innerhalb EMSG-Master auf einer eigenen Maske. Die Darstellung erfolgt dabei in Tabellenform.

Die Liste der auslösenden Aktionen ist in den Systemanforderungen [7] definiert.

### Überwachung

Für EMSG wird keine Schnittstelle zur Überwachung realisiert.

### Fehlerbehandlung

Hier wird zwischen erwarteten und unerwarteten Fehlern unterschieden. Erwartete Fehler (z.B. Validierungsfehler, siehe Abschnitt 4.8.2) werden vom System vollständig behandelt. Beim Auftreten wird auf der Benutzeroberfläche eine Fehlermeldung ausgegeben. Diese Fehler werden in der vom Benutzer im Browser eingestellten Sprache ausgegeben.

Unerwartete Fehler (z.B. Programmabsturz durch Programmierfehler) hingegen führen dazu, dass die aktuell laufende Aktion unterbrochen wird. Diese Fehler werden in das Anwendungslog (siehe Abschnitt 4.8.3) protokolliert. Auf der Benutzeroberfläche wird in diesem Fall eine allgemeine Fehlerseite weitergeleitet. Detaillierte Informationen zum Fehler (wie z.B. Stacktrace) werden aus Sicherheitsgründen nicht ausgegeben. Diese und weiterführende Informationen können dem Anwendungslog entnommen werden.

### Mehrsprachigkeit

Die Mehrsprachigkeit wird mittels des Ressourcen basierenden Lösungsansatzes von Microsoft realisiert. Es werden die Sprachen Deutsch, Französisch und Italienisch unterstützt. Pro Sprache gibt es mehrere Ressource-Dateien zwecks besserer Übersichtlichkeit und Wartbarkeit. Sprachübergreifend haben alle Ressource Dateien den gleichen Aufbau.

#### Aufteilung in mehrere Ressource-Dateien

Folgende Tabelle zeigt die Gliederung der Übersetzungen in mehrere, einzelne Dateien. Pro Sprache gibt es jede Datei mit ihren eigenen Übersetzungen.

| **Ressource-Datei** | **Inhalt** | **Beispieltexte (Deutsch)** | **Master** | **Mobile** |
| --- | --- | --- | --- | --- |
| EditorLocalization.resx | Allgemeine Text für Detailformulare  (Telerik) |  | x |  |
| GridLocalization.resx | Allgemeine Texte für Tabellen  (Telerik) | „Aufsteigend sortiert“ „Keine Daten“ | X |  |
| UploadLocalization.resx | Allgemeine Texte fürs Hochladen  von Dateien  (Telerik) |  | X |  |
| ButtonLocalization.resx | Texte von Schaltflächen | „Check-Out rückgängig machne“ | X |  |
| EnumLocalization.resx | Texte von statischen Wertevorräten  (Katalog) | „Asphalt“ (Belagstyp) | X |  |
| GridHeaderFooterLocalization.resx | Überschriften von Tabellen |  | X |  |
| LookupsLocalization.resx | Texte von dynamischen Wertevorräten (Katalog) | „IA“ (Belastungskategorie) | X |  |
| MapLocalization.resx | Texte die auf der Karte angezeigt werden | „Gebäude“ | X |  |
| MenuLocalization.resx | Texte für Menüeinträge | „Menge pro Belastungskategorie“ | X |  |
| ModelLocalization.resx | Labels von den Eingabeformularen (Masken) | „Strassenname“  (Strassenabschnitt) | x |  |
| NotificationLocalization.resx | Erfolgsmeldungen, Fehlermeldungen, Bestätigungsmeldungen von Popups | „Das Achsenupdate wurde erfolgreich durchgeführt“  „Wollen Sie den Zustandsabschnitt wirklich löschen?“ | X |  |
| ReportLocalization.resx | Texte für Auswertungen von Microsoft Report Assemblies | „WBW [Mio. CHF] WV [kCHF]“ | X |  |
| TextLocalization.resx | Allgemeine Texte | „Aktueller Mandant“ | X |  |
| TitleLocalization.resx | Titel der einzelnen Seiten | „Zustände und Massnahmenvorschläge“ | X |  |
| ValidationErrorLocalization.resx | Fehlertexte von strukturellen  Validierungen |  | X |  |
| MobileLocalization.resx | Alle Texte |  |  | X |

Tabelle 8: Gliederung Ressource-Dateien

Jede Ressource-Datei existiert für jede Sprache einmal. Darüber hinaus existiert für jede Ressource-Datei eine Default-Datei. Diese kann nicht editiert werden. Auf die Default-Datei wird zurückgegriffen falls durch eine fehlerhafte durchgeführte Änderung (siehe dazu 4.8.7.2 und 4.8.7.3) der gesuchte Key nicht gefunden werden konnte. Die Default-Datei gewährt somit ein stabiles Systemverhalten.

#### Änderungen vorbereiten und durchführen

Die aktuellen Ressource-Dateien von EMSG (sowohl EMSG-Master und EMSG-Mobile) können in EMSG-Master als Archiv (ZIP) heruntergeladen werden. Ausgenommen davon sind die Default-Dateien da diese nicht geändert werden können.

Änderungen in den Ressource-Dateien werden lokal ausserhalb des Systems vorgenommen. Das Bearbeiten der Ressource-Dateien kann mit einem herkömmlichen Texteditor vorgenommen werden. Darüber hinaus gibt es eine Reihe von spezialisierten Editoren, die das Bearbeiten vereinfachen (z.B. Resx Localization Studio[[31]](#footnote-32)).

#### Änderungen bereitstellen

Nach dem Bearbeiten der Ressource-Dateien können diese einzeln über eine eigene Webseite in EMSG-Master hochgeladen werden. Ab dann sind die Änderungen für EMSG-Master sofort aktiv. Änderungen sollen nur im Wartungsfenster hochgeladen werden da durch das Speichern einer Ressource Datei am Applikationsserver der IIS-Application-Pool neu gestartet wird. Änderungen für EMSG-Mobile werden mit dem nächsten Check-Out aktiv (siehe 5.1.2.1).

EMSG-Master führt beim Hochladen eine Validierung der Ressource-Datei durch. Falls die Validierung fehlt schlägt wird ein Fehler ausgegeben. In diesem Fall wird die Ressource-Datei nicht übernommen.

Es kann immer nur eine Ressource-Datei hochgeladen werden. Mehrere Ressource-Dateien können nicht gleichzeitig hochgeladen werden, da sich durch das Überschreiben einer Ressource-Datei der IIS-Application-Pool recycled.

#### Beispiel: Inhalt von EnumLocalization.resx

Abbildung 58 zeigt beispielhaft, wie die Ressource-Dateien inhaltlich aufgebaut sind.

|  |
| --- |
| <data name="BelagsTyp\_Asphalt" xml:space="preserve">  <value>Asphalt</value>  </data>  <data name="BelagsTyp\_Beton" xml:space="preserve">  <value>Beton</value>  </data>  <data name="EigentuemerTyp\_Gemeinde" xml:space="preserve">  <value>Gemeinde</value>  </data>  <data name="EigentuemerTyp\_Korporation" xml:space="preserve">  <value>Korporation</value>  </data>  <data name="EigentuemerTyp\_Private" xml:space="preserve">  <value>Private</value>  </data> |

Abbildung 58: Inhalt Ressource Datei

#### Auswahl der Sprache

Beim erstmaligen Einstieg in EMSG-Master wird die Sprache Deutsch (de-ch) gewählt. Innerhalb von ESMG-Master besteht die Möglichkeit die Sprache zu ändern. Neben Deutsch steht Italienisch (it-ch) und Französisch (fr-ch) zur Verfügung. Die gewählte Sprache wird am Client in Form eines Cookie gespeichert. Beim erneuten Einstieg in EMSG-Master wird die gewählte Sprache aus dem Cookie übernommen.

### Hilfe

Die Context bezogene Online Hilfe wird in statischen HTML realisiert. Es werden die Sprachen Deutsch, Französisch und Italienisch unterstützt. Der Inhalt der Online Hilfe wird in EMSG-Master in einem eigenen Browserfenster dargestellt. In EMSG-Mobile wird die Online Hilfe in einem WPF Control dargestellt.

#### Aufteilung in mehrere HTML Dateien

Die Context bezogene Online Hilfe wird für jede Seite/Screen durch eine eigene HTML Datei repräsentiert. Der Aufruf der Online Hilfe erfolgt über ein entsprechendes Icon. Falls es keine entsprechende Online Hilfe gibt wird das Icon nicht angezeigt.

#### Änderungen vorbereiten und durchführen

Die Online Hilfe für EMSG-Master und EMSG-Mobile wird separat in Form eines Archiv (zip) von EMSG-Master heruntergeladen.

In dem Archiv befinden sich alle unterstützen Sprachen. Die Änderungen in HTML können beispielsweise mit einem herkömmlichen Texteditor[[32]](#footnote-33) oder Microsoft Word durchgeführt werden.

Für EMSG-Master kann der Umfang der Hilfe angepasst werden (z.B. Neue HTML Dateien hinzufügen, bestehende HTML Dateien entfernen). In EMSG-Mobile gibt es nur wenige Screens (ca. 4). Der Umfang der Hilfe kann hier nicht angepasst werden.

#### Änderungen bereitstellen

Die Online Hilfe für EMSG-Master bzw. EMSG-Mobile wird als Archive (ZIP) in EMSG-Master hochgeladen. Die existierende Online Hilfe wird gelöscht und durch die neue komplett ersetzt.

In EMSG-Master wird die neue Online Hilfe sofort aktiv. In EMSG-Mobile wird sie mit dem nächsten Check-Out aktiv (siehe 5.1.2.1).

#### Aufbau des Archiv und Mapping in EMSG-Master

Der Name des Archiv lautet MasterHelpSystem.zip. Für jede unterstütze Sprache gibt es in dem Archiv je ein Verzeichnis (de, fr, it).

Das Mapping einer HTML Datei erfolgt über die URL und den Namen der View in MVC. Das Mapping wird an Hand eines Beispiels im Strassennamen Modus für „Netzdefinition“ (Liste der Strassenabschnitte und Editieren eines Strassenabschnitt) illustriert.

**Beispiel: Liste der Strassenabschnitte**

|  |  |
| --- | --- |
| **Sprache** | de |
| **URL** | http(s)://<Server>/NetzverwaltungStrassennamen/NetzdefinitionUndStrassenabschnitt |
| **View** | Index.cshtml |
| **HTML Datei** | de/NetzverwaltungStrassennamen/NetzdefinitionUndStrassenabschnitt\_index.htm |

**Beispiel: Editieren eines Strassenabschnitts**

|  |  |
| --- | --- |
| **Sprache** | de |
| **URL** | http(s)://<Server>/NetzverwaltungStrassennamen/NetzdefinitionUndStrassenabschnitt |
| **View** | EditStrassenabschnitt.cshtml |
| **HTML Datei** | de/NetzverwaltungStrassennamen/NetzdefinitionUndStrassenabschnitt\_ EditStrassenabschnitt.htm |

#### Aufbau des Archiv und Mapping in EMSG-Mobile

Der Name des Archiv lautet MobileHelpSystem.zip. Für jede unterstütze Sprache gibt es in dem Archiv je ein Verzeichnis (de, fr, it).

Für die 4 Screens stehen folgende HTML Dateien zur Verfügung:

* ZustandsabschnittForm.html
* ZustandsabschnittMap.html
* ZustandsabschnittTrottoirZustand.html
* ZustandsabschnittFahrbahnZustand.html

## Design-Strategien EMSG-Mobile

### Locking

Es findet kein Datenlocking statt, die Daten werden nur auf dieser Maschine bearbeitet. Während der Bearbeitungszeit ist der Datensatz in der Datenbank des EMSG Master mit einem CheckOut versehen und wird via EMSG Master Geschäftslogik vom Bearbeiten gesperrt. Dieser Lock ist als eigener Datensatz zu verstehen und nicht als ein in der Datenbank implementierter Zugriffslock.

### Validierung

Diese Validierung der Objekte erfolgt bereits in der Präsentationsschicht. Die Validierung erfolgt dabei im Ersten Schritt in der Benutzeroberfläche durch die zugehörigen ViewModels. Im zweiten Schritt erfolgt die Validierung in der Geschäftslogik beim Model Binding.

Folgende strukturelle Validierungen werden auf den Objekten durchgeführt:

* Einhaltung der Wertebereiche (z.B. Bereich, Min, Max, etc.)
* Pflichtfelder
* Einhaltung des Datentyp

### Logging

Im Anwendungslog (oft auch als technisches Log bezeichnet) werden unerwartete Fehler geloggt. Diese Informationen werden dabei in einer Textdatei am Applikationsserver gespeichert. Pro Logeintrag werden folgenden Informationen gespeichert:

* Windows User Name
* Zeitstempel
* Fehlertext
* Stacktrace
* Fehlerverfolgungsnummer

Falls ein unerwarteter Fehler auftritt, wird am Client eine Fehlermeldung dargestellt. Als Komponente zum Loggen wird NLog (siehe Abschnitt 4.6.1.2) eingesetzt.

### Überwachen

Für EMSG wird keine Schnittstelle zur Überwachung realisiert.

### Fehlerbehandlung

Hier wird zwischen erwarteten und unerwarteten Fehlern unterschieden. Erwartete Fehler (z.B. Validierungsfehler, siehe Abschnitt 4.9.24.8.2) werden vom System vollständig behandelt. Beim Auftreten wird auf der Benutzeroberfläche eine Fehlermeldung ausgegeben. Diese Fehler werden in der jeweiligen Sprache ausgegeben.

Unerwartete Fehler (z.B. Programmabsturz durch Programmierfehler) hingegen führen dazu, dass die aktuell laufende Aktion unterbrochen wird. Diese Fehler werden in das Anwendungslog (siehe Abschnitt 4.8.3) protokolliert. Der Benutzer erhält in diesem Fall eine allgemeine Fehlermeldung. Detaillierte Informationen zum Fehler (wie z.B. Stacktrace) werden aus Sicherheitsgründen nicht ausgegeben. Diese und weiterführende Informationen können dem Anwendungslog entnommen werden.

### Mehrsprachigkeit

Durch die Installation von EMSG-Mobile wird die Sprache konfiguriert. Änderungen der Sprache zur Laufzeit ist nicht vorgesehen. Die Sprachübersetzungen erfolgen mittels Ressource-Dateien (siehe 4.8.7).

### Hilfe

Der EMSG-Mobile Client wird während der Installation auf eine Sprache festgelegt, Änderungen der Lokalisierung während der Laufzeit sind nicht vorgesehen. Die Sprachübersetzungen erfolgen mittels HTML-Dateien (siehe 4.8.8).

# Schnittstellen

Dieses Kapitel beschreibt die wesentlichen internen und externen Schnittstellen des Systems EMSG. Abbildung 59 gibt eine Übersicht über die Schnittstellen und die verwendeten Protokolle.

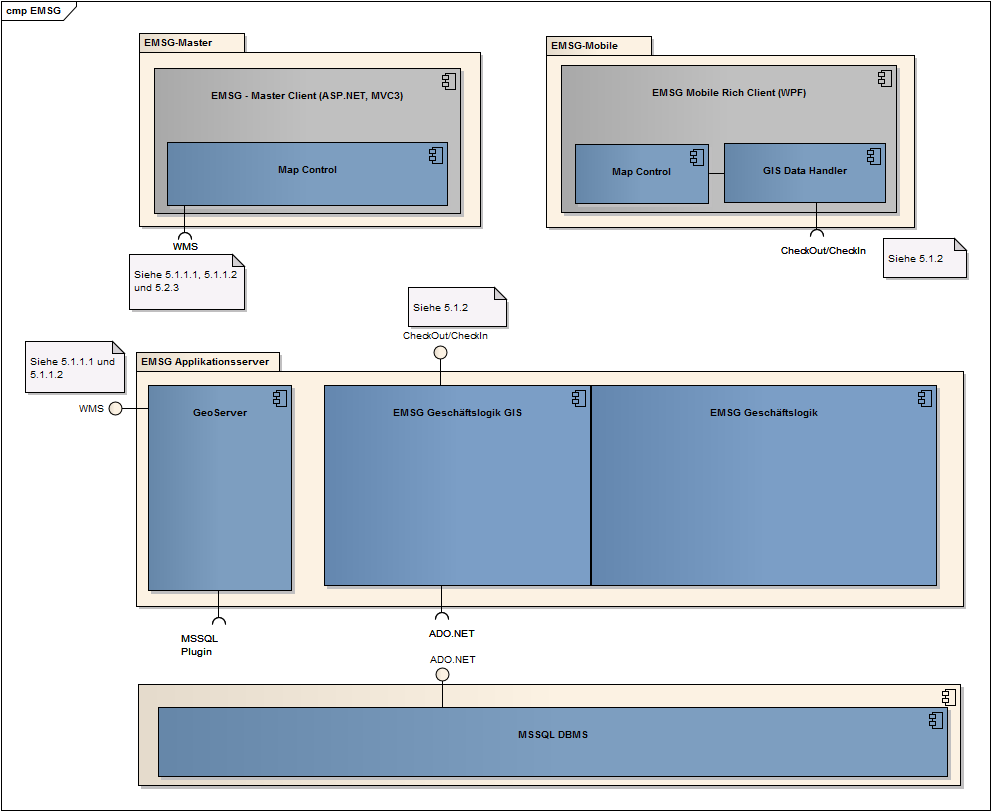


Abbildung 59: Schnittstellen

## Interne Schnittstellen

Dieser Abschnitt beschreibt die sich aus der Architektur ergebenden internen Schnittstellen zwischen den EMSG-Subsystemen.

### Karte (GeoServer)

#### Karten-Zoomlevels

Das EMSG-Master (Web) Kartenelement stellt die GIS-Layer in definierten Zoomstufen zu Verfügung. Diese Zoomstufen sind in Hinsicht auf Performance und Übersichtlichkeit der Bearbeitung optimiert.

Die Zoomstufen sind über die Systemkonfigurationsdatei „web.config“ frei konfigurierbar. Details können dem EMSG Konfigurationshandbuch web.config [11] entnommen werden.

#### Kartenauswertungen (WMS-T)

Die Auswertungen können in der Karte dargestellt werden. Alle GIS-Auswertungen werden vom GeoServer erzeugt und greifen im Standardfall immer auf das aktuelle Jahr zu. Folgende Einschränkungen stellen die Services zur Verfügung:

* **Zeitliche Einschränkung**  
  Zeitliche Einschränkungen werden über die Erweiterung von WMS – WMS-T (WebMapService – Time) implementiert. Diese erlaubt es, die zeitliche Dimension bei der Abfrage zu berücksichtigen.
* **Attributive Einschränkung**  
  Attributive Einschränkungen werden in Form von definierten Layer implementiert. Der Wertebereich muss bekannt und statisch vorliegen. Die Einschränkung kann demnach durch korrekte Auswahl der Layer erfolgen (für den EMSG Benutzer bleibt diese Logik verborgen; das System übersetzt die Einschränkungen automatisch in die zu ladenden WMS-Layer).

Welche Auswertungen (Karten) im GIS-Modus zur Verfügung stehen, ist in den Systemanforderungen [7] beschrieben.

##### Datengrundlage Kartenauswertungen

Kartenauswertungen im EMSG-Master, bzw. die Visualisierung basiert immer auf den in EMSG Fachdatensatz enthaltenen Geometrien. Diese Geometrien entsprechen dem letzten Stand der Bearbeitung des Datensatzes[[33]](#footnote-34). Wesentlich ist, dass diese Geometrien nicht dem aktuellen Stand der Achsdaten entsprechen müssen.

Werden Achsdaten visualisiert, so wird immer auf den aktuellsten Stand der Achsen zurückgegriffen. Dies bedeutet, dass es potentiell Lageunterscheide zwischen älteren EMSG Fachdaten und aktuellen EMSG Achsen kommen kann.

Da die Auswertungen und deren Funktionalität auch im Konzept für den Multimonitorbetrieb [8] beschrieben werden, wird dieser Punkt auch noch im Zuge der Detaillierung des Multimonitorbetriebes näher behandelt.

##### MapProxy

MapProxy wird als Vereinheitlichungselement eingesetzt. Services und Layer die nur als WMTS vorhanden sind können durch MapProxy als WMS für die Auswertungen zur Verfügung gestellt werden.

##### Achsenimport

Neben der Möglichkeit Achsen über die Interlis 2 Schnittstelle zu importieren ist auch der Import von shapefiles möglich. Neben dem shapefile (.shp), das die Geometrien beinhaltet, wird auch ein database file (.dbf) für den Import benötigt. Im database file muss das Attribut des Achsennamens (name) definiert sein.

Beide files sind erforderlich, um Achsen erfolgeich in die Datenbank importieren zu können.

### Datenaustausch EMSG-Master mit EMSG-Mobile

Um ein offline Arbeiten auf dem EMSG-Mobile Client zu ermöglichen, müssen die relevanten Daten von einer oder mehreren Inspektionsrouten auf das EMSG-Mobile Gerät „ausgecheckt“ werden können. Die ausgecheckten Daten werden in der EMSG-Master Datenbank entsprechende markiert, und es wird bei einer Bearbeitung durch einen EMSG-Master Benutzer eine entsprechende Fehlermeldung ausgeben.

Nach der Zusammenstellung seiner Inspektionsrouten auf dem EMSG-Master kann der Benutzer entsprechende Routen auswählen und als Datenpakete auf dem Server zusammenstellen. Dabei müssen auch die entsprechenden Hintergrundkarten in mehreren Auflösungen in dem Datenpaket inkludiert werden. Das Datenpaket wird dann vom Nutzer heruntergeladen und z.B. über einen USB-Stick auf das EMSG-Mobile Gerät transferiert.

Nach der Bearbeitung der Daten über den EMSG-Mobile Client transferiert der Nutzer die Daten wieder auf den EMSG-Master, indem dort ein Import gestartet wird.

**Umfang des Datenpaketes** (für eine Übersicht siehe Abschnitt 4.5.2):

* Aktuelle EMSG Fachdaten (inkl. Zustandsprotokolle), EMSG Referenzen und EMSG Achsen, welche Teil der Inspektionsrouten sind, (Vektorgeometrien) lt. EMSG-Mobile Datenmodell (siehe Abschnitt 4.4.9.1)
* Rasterdaten (siehe Abschnitt 4.4.9.2)
* Sprachübersetzungen und Online Hilfe

#### Check-Out (EMSG-Master 🡪 EMSG-Mobile)



Abbildung 60: Aktivitätsdiagramm Check-Out von Daten für die Bearbeitung am EMSG-Mobile (Download des Datenpaketes am EMSG-Master)

Für das Check-Out der Daten vor der Übertragung zu EMSG-Mobile wählt der Anwender eine oder mehrere Inspektionsrouten aus (siehe auch Abbildung 60). Das entsprechende Paket wird vom EMSG-Server zusammengestellt und kann danach heruntergeladen werden. Dieser Prozess arbeitet synchron – ein Weiterarbeiten während das Paket zusammengestellt wird ist deswegen nicht möglich. Als Ergebnis (Download über Browser) erhält der EMSG Anwender ein Datenpaket in Form eines Zip-Archivs, welches er nun auch offline (z.B. mit USB Memory-Stick) zum EMSG-Mobile Tablet PC transferieren kann.

Datensätze, welche im EMSG-Master als „CheckedOut“ markiert sind, können dort nicht bearbeitet oder gelöscht werden. Diese Datensätze sind für die Benutzer gesperrt.

Um das heruntergeladene Datenpaket im EMSG-Mobile bearbeiten zu können, muss dieses in die EMSG-Mobile Applikation importiert werden (siehe Abbildung 61). Wesentlich ist, dass eine EMSG-Mobile Anwendung immer nur ein Datenpaket bearbeiten kann. Durch den Import des Datenpakets werden ebenfalls die Sprachübersetzungen und die Online Hilfe aktualisiert. Des Weiteren wird während des Importprozesses geprüft, ob die Version des Datenpaketes kompatibel[[34]](#footnote-35) mit der Version der installierten EMSG-Mobile Anwendung ist. Folgende Validierungen finden während des Importprozesses statt:

* Versionsvergleich: Die Datenpaketversion wird mit der EMSG-Mobile Version verglichen. Sind die Versionen nicht kompatibel, so kann das Datenpaket nicht importiert werden. Der Anwender erhält eine Meldung, dass er seine EMSG-Mobile Anwendung aktualisieren muss. Wesentlich ist, dass das Datenpaket die Information beinhaltet mit welchen EMSG-Mobile Software-Versionen es kompatibel ist.
* Prüfung auf offene Bearbeitung eines CheckOuts am EMSG-Mobile: Der Anwender kann vorhandene Bearbeitungen durch Bestätigung überschreiben.



Abbildung 61: Aktivitätsdiagramm Check-Out von Daten für die Bearbeitung am EMSG-Mobile (Importieren des Datenpaketes in EMSG-Mobile)

#### Check-In (EMSG-Mobile 🡪 EMSG-Master)



Abbildung 62: Aktivitätsdiagramm Check-In von Daten aus dem EMSG-Mobile in die EMSG-Master Anwendung

In der EMSG-Mobile Applikation werden alle Änderungen, die der Bearbeiter durchführt in einer eigenen Tabelle (Änderungsdatei) mitgeführt. Diese Tabelle beinhaltet für jede Änderung und Datensatz die entsprechende Aktion:

* Create
* Update
* Delete

Mit Beendigung des Bearbeitungsvorgangs[[35]](#footnote-36) besteht für den Anwender die Möglichkeit, eine Änderungsdatei zu exportieren und diese auf einen geeigneten Arbeitsplatz (manuell, z.B. via USB Memory-Stick) zu übertragen. Nach dem Export der Änderungen können die ausgecheckten Daten nicht mehr in EMSG-Mobile bearbeitet werden. Der Datensatz wird automatisch durch die EMSG-Mobile Anwendung nach erfolgreichem Export inaktiv gesetzt (siehe auch Abbildung 62). Die Datei kann dann auf den EMSG-Master Server hochgeladen und die Änderungen in der EMSG-Master Datenbank durchgeführt werden. Der Import-Vorgang arbeitet synchron. Mit Abschluss des Vorgangs werden die entsprechenden Datensätze wieder auf „CheckedIn“ gesetzt und können in Folge auch am EMSG-Master wieder bearbeitet werden..

Jeder „CheckOut“ hat eine Versionsnummer, welcher angibt mit welcher Softwareversion dieser „CheckOut“ erstellt wurde. Diese Versionsnummer ist auch in der Änderungsdatei enthalten und wird beim „CheckIn“ überprüft. Prinzipiell ist der „CheckIn“ abwärtskompatibel. Dies bedeutet, dass alte Änderungsdateien prinzipiell durch den Anwender eingecheckt werden können. Im EMSG-Master wird durch die RE konfiguriert, bis zur welchen Version diese Abwärtskompatibilität gewährleistet wird. Versionen, welche eine niedrigere Version aufweisen, können nicht mehr eingecheckt werden. Der Anwender erhält dann eine entsprechende Fehlermeldung. Dieser Ansatz erlaubt es, bei potentiell grösseren strukturellen Softwareänderungen sicherzustellen, dass nur gültige Änderungsdateien im EMSG-Master importiert werden können.

#### Check-Out Aufheben

Im EMSG-Master steht dem Benutzeradministrator auch die Funktionalität zur Verfügung ein Check-Out aufzuheben – damit werden die entsprechenden Datensätze wieder für die Bearbeitung im EMSG-Master freigeschalten. Ein Check-In für einen bereits aufgehobenen Check-Out ist nicht mehr möglich.

## Externe Schnittstellen

Dieser Abschnitt beschreibt die sich aus dem Pflichtenheft [2] ergebenden externen Schnittstellen zu anderen Systemen, sowohl innerhalb als auch ausserhalb des MISTRA-Systems.

Abbildung 63 zeigt die Datenflüsse zwischen den internen und externen Schnittstellen.

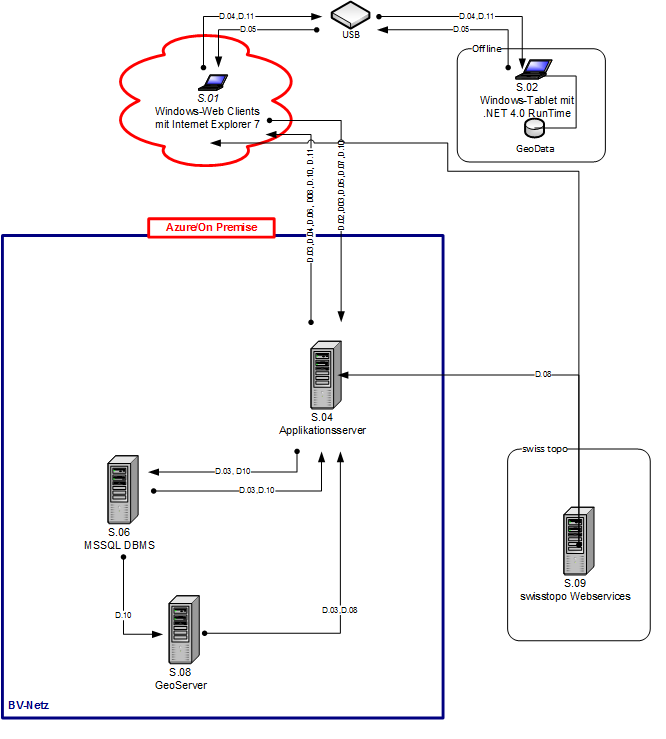


Abbildung 63: Datenflussskizze

### MISTRA Basissystem

#### INTERLIS 2

Siehe Abschnitt 4.3.2 für Details zum Bezug der Achsen über die Offline-Schnittstelle INTERLIS 2. Der Prozess für den Bezug der Daten über die INTERLIS 2 Schnittstelle ist asynchron. Die Datenpakete werden im Verzeichnis „In“ des AchsenImportServices abgelegt (siehe Installationshandbuch) und von dort in die Datenbank importiert. Die Mengengrenze liegt bei rund 1 GB pro Inkrementdatei und 3.5 GB für den Initialimport. Die inkrementellen Importdateien können auch nach Mandant getrennt werden um die maximale Grösse von rund 1 GB zu unterschreiten. Sollte diese Massnahme nicht ausreichen können die Inkrementdateien auch noch weiter getrennt werden.

### WMS – swisstopo

Swisstopo stellt performante GIS-Hintergrundlayer zur Verfügung, die als Web Map Service (OGC Standard, Version 1.3.0) im Client eingebunden werden. Alle WMS-Services werden zwecks client-seitigem Caching als Tiled-Layer eingebunden. Dies hat den Vorteil, dass der Anwender schneller erste Teileausschnitte der Karten erhält und bei geringfügigen „Pan“-Aktionen nur wenige Daten geladen werden müssen. In Summe reduziert der Tiled-WMS Ansatz die Datenmenge, die zwischen swisstopo und dem EMSG-Master Webclient übertragen werden. Die Anzahl der Requests bei Tiled-Layer Abfragen ist höher als bei Single-Tiled Layern. Alle Daten werden in der Projektion EPSG:21781 bezogen.

Der Zugriff auf das entsprechende Service erfolgt über die EMSG-Master Serveranwendung, die für autorisierte Anwender einen Proxy zur Verfügung stellt. Die EMSG-Master Anwendung fügt aber weder Caching-Funktionalität noch funktionale Erweiterungen mit Ausnahme der Autorisierung hinzu.

Layer, die von swisstopo eingebunden werden, sind nicht mandantenabhängig und haben damit keine eingeschränkte Sichtbarkeit.

Folgende swisstopo WMS Layer stehen am EMSG-Master Webclient zur Verfügung[[36]](#footnote-37):

* Orthophotos
* Karte Farbig
* Karte Schwarz/Weiss
* Zusatzinformationen
  + Hausnummern
  + Strassenlärm bei Tag
  + Strassenlärm bei Nacht
  + Bahnlärm bei Tag
  + Bahnlärm bei Nacht
  + Grundwasservorkommen
  + Grundwasservulnerabilität
  + Kataster Belasteter Standorte

### WMTS – swisstopo

WMTS Dienste sind eine performantere Alternative zu den WMS Diensten. Die WMTS Dienste werden für die Hintergrundkarten als auch für die AV Daten (sofern von swisstopo angeboten) verwendet.

Alle Daten werden in der Projektion EPSG:21781 bezogen, sind aber in den Zoomstufen begrenzt (WMTS wird mit vorberechneten Kacheln betrieben, der Server liefert nur die vorhandenen Zoomstufen aus). Aus diesem Grund kann nicht gewährleistet werden, dass jeder Layer in allen Zoomstufen vorhanden ist oder in der entsprechenden Qualität vorliegt.

Folgende WMTS Layer stehen am EMSG-Master zur Verfügung:

* + AV Daten: Zusammenfassung der oben genannten Layer durch swisstopo. Der Inhalt der Daten kann von EMSG nicht beeinflusst werden.
    - OSNR: Liegschaftsnummern
    - LOCPOS: Strassennamen
    - LNNA: Nomenklatur
    - RESF: Liegenschaften
    - LCSFC: BoFläche (farbig)
  + Orthofotos
  + Karte farbig
  + Karte schwarz/weiß
  + Zusatzinformationen
    - Hausnummern
    - Strassenlärm bei Tag
    - Strassenlärm bei Nacht
    - Bahnlärm bei Tag
    - Bahnlärm bei Nacht
    - Grundwasservorkommen
    - Grundwasservulnerabilität
    - Kataster Belasteter Standorte

Die Konfiguration ob WMS oder WMTS verwendet werden soll sowie der URLs der Dienste erfolgt in der web.config des EMSG-Master Services (siehe [11]).

### Mengengerüst GIS-Schnittstellen

Tabelle 9 dokumentiert die zu erwartende und durch die EMSG Fachapplikation nach den Performancekriterien laut Pflichtenheft [2] zu verarbeitenden Datenmengen:

|  |  |  |
| --- | --- | --- |
| **Schnittstelle** | **Mengengerüst** | **Anmerkung** |
| INTERLIS 2 | ca. 10 bis 15 MB pro Mandant |  |
| WMS swisstopo (EMSG-Master) | Annahme: 500 Kacheln pro Bearbeitungsphase, ca. 10-20 Bearbeitungsphasen pro Jahr.  5‘000 – 10‘000 Kacheln pro Mandant pro Jahr. |  |
| WMS GeoServer | Ähnliche wie WMS swisstopo, WMS muss aber nach GIS - Bearbeitung aktualisiert werden:  ca. 50‘000 Kacheln pro Mandant pro Jahr |  |
| Check-Out EMSG-Master 🡪 EMSG-Mobile | Annahme: Mandaten – räumlicher Bereich ca. 5 km² , Zwei Layer (Raster - swisstopo) werden offline vorgehalten, ca. 4 Checkouts pro Mandant und Jahr  Ca. 10 MB pro Mandant und Check-Out  Ca. 40 MB pro Mandant und Jahr |  |

Tabelle 9: Mengengerüst GIS-Schnittstellen

# Anforderungszuordnung

Dieses Kapitel aus der HERMES-Vorlage entfällt durch Tailoring, da es nicht zur ordnungsgemässen Dokumentation benötigt wird.

# Realisierbarkeitsuntersuchungen

Die Realisierbarkeit der Anwendung EMSG wurde durch Erstellen eines Prototyps bzw. durch gesammelte technische Erfahrungen aus bereits abgeschlossenen Produkten geprüft.

Grundsätzlich wird in der Realisierung so vorgegangen, dass zu Beginn der Realisierung die verschiedenen Schnittstellen und Technologien zumindest einmal verwendet und eingesetzt werden um frühzeitig etwaige Probleme zu erkennen. In der späteren Realisierungsphase werden dann zu diesen „Durchstichen“ die weiteren Varianten umgesetzt.

## Prototypen

Um eine optimale Realisierung der Fachapplikation EMSG zu gewährleisten wurden folgende Komponenten in Form eines Prototyps untersucht und evaluiert. Das Augenmerk lag hier vor allem auf das Zusammenspiel dieser Komponenten:

* OpenLayers
* MVC3
* NHibernate
* .NET Topology Suite
* Oracle DBMS

Als Ergebnis des Prototyps lag eine simple Webapplikation vor, welche einen einfachen allgemeinen GIS Bearbeitungsfall[[37]](#footnote-38) abbildet und damit das erfolgreiche Zusammenwirken der aufgeführten Komponenten bestätigt. Als besonders risikoreich wurde das Zusammenspiel der GIS-Komponenten, des O/R-Mappers und der Datenbank gesehen. Dieses Zusammenspiel konnte durch den Prototypen erfolgreiche bestätigt werden.

* + 1. **Achsenaktualisierung**

In Iteration 1 wurden erste Tests zum komplexen Abgleich der Achsen zwischen dem Basissystem und der Fachapplikation EMSG durchgeführt. Grundlage für die Tests und die Herangehensweise wurden bereits im Kapitel 4.3.2 dokumentiert.

**Ausgangslage**

Folgende Datensätze wurden für die Tests herangezogen:

* Full Export INTERLIS 2 (Gesamte Schweiz): 20111213110540\_Axis\_CH\_Kt\_FULL\_2011\_09\_1.xtf (ca. 148000 Datensätze)
* Inkrementeller INTERLIS 2 Export (Gesamte Schweiz): 2012\_01\_04\_export\_Achsen\_CH\_incr\_1.09.2011\_19.12.2011.xtf

Weiters wurden folgende Services des Basissystems verwendet: (Zugang über RAS Wartung)

* TransformationService
* LineareferenceServcie
* HistoryService

**Entwicklung:**

Folgende Komponenten wurden für die Durchführung der Tests entwickelt. Der Code steht in einem eigenen Branch zur Verfügung und stellt die Grundlage für die weitere Implementierung dar:

* Parser für INTERLIS 2 Daten (Full und Inkrementell)
* Implementierung SOAP Schnittstellen (LinearReferenceService, TransformationService, HistoryService)
* Importprozedur von INTERLIS 2 Daten in ein OracleDBMS

**Durchgeführte Tests:**

|  |  |  |
| --- | --- | --- |
| **Testbeschreibung** | **Ergebnis** | **Kommentar** |
| Parsen des Full INTERLIS Exports und Import in ein OracleDBMS | Test erfolgreich. Dauer ca. 1 Stunde | Für das Parsen der INTERLIS 2 Daten wurde eine Kombination von XmlReader und XmlSerializer eingesetzt – diese erlaubt ein arbeitsspeicherschonendes Einlesen der Daten. |
| Parsen des Inkrementellen Exports und Aktualisierung der Achsdaten in einem OracleDBMS | Test erfolgreich | Datenmenge der Inkrementellen INTERLIS 2 Daten nicht aussagekräftig für Abschätzung der Prozessdauer |
| Zugriff auf HistoryService über SOAP | Test erfolgreich |  |
| Zugriff auf LinearReferenceService über SOAP | Test erfolgreich |  |
| Zugriff auf TransformationService über SOAP | Dauer: TransformXyToRBBS ~ 1 Sekunde |  |

**Erkenntnisse:**

* Erwartende Datenmengen der INTERLIS 2 Daten sind kein Problem für den Abgleich der Achsdaten
* Services des Basissystems liefern Antworten in akzeptablen Zeiten
* Verwendung des HistoryServices aus Sicht der RE nicht notwendig, da die Inhaltliche Logik bereits in „UpdateLinearStable“ vorhanden ist.
* Verwendung von UpdateLinearStable sinnvoll
* TransformXyToKM liefert keine Kilometerwerte für Achsen von Gemeinden.

**Umsetzungsempfehlungen:**

* „Kopie Achsdaten“ stellt immer nur den aktuellen Stand der Achsen dar und beinhaltet nicht den vollständigen Versionsverlauf. Es werden aber auch Löscheinträge gespeichert. Eine vollständige Versionierung der „Kopie Achsdaten“ ist nicht notwendig,
* EMSG Achsdaten liegen versioniert vor. Des Weiteren werden EMSG Achsdaten einem Jahresabschluss zugeordnet.
* Einsatz des „HistoryServices“ nicht notwendig. Da ChangeCode keine sinnvolle Aussage für GIS Laien liefern und die wesentliche Logik bereits im UpdateLinearStable Service enthalten ist.

## Referenzprodukte

Dieser Abschnitt listet Referenzprodukte, in denen der Auftragnehmer die für EMSG vorgesehenen Komponenten in ähnlichen Problembereichen schon erfolgreich eingesetzt hat.

### Geoshop

**Referenz**: Web-Anwendung mit integrierter Kartenfunktion

**Beschreibung**: Dabei handelt es sich um eine Webanwendung zur Abgabe von Geodaten des Landes Niederösterreich (NÖL). Mitarbeiter der NÖL, sowie externe Benutzer (Firmen, öffentliche Einrichtungen, Privatbenutzer, etc.) können Interessensgebiete über eine dynamische Karte auswählen und daraufhin Geodaten Produkte aus diesem Gebiet erwerben. Diese werden dann in passenden Formaten (DVD, ausgedruckt, als Download, etc.) versendet.

**Eingesetzte Komponenten und Technologien:**

* Microsoft Visual Studio 2008
* Resharper
* Unit Test Framework
* C# 3.5
* Microsoft IIS
* ASP.NET
* MVC
* JavaScript
* JQuery
* ArcGIS

### JaNo

**Referenz**: Smart Client mit integrierten Auswertungen

**Beschreibung**: JaNo ist das Landeslehrerverwaltungs- und Abrechnungsprogramm, mit dessen Hilfe der Landesschulrat für Niederösterreich und seine nachgeordneten Dienststellen und Strukturen (Schuldirektionen, Bezirksschulräte, regionales Bildungsmanagement) folgende Aufgaben abwickelt:

* Planstellenverbrauchsermittlung
* Begleitung des Verfahrens zum Landeslehrercontrolling
* Kontrolle und Abwicklung von Bestandteilen der Lehrerbesoldung, wie EMDL, MDL, Abgeltungen, Zulagen und Vergütungen
* Erstellen von Auswertungen und Statistiken zum Unterrichtseinsatz der LandeslehrerInnen

**Eingesetzte Komponenten und Technologien:**

* Visual Studio 2010
* Resharper
* SpecFlow
* Unit Test Framework
* C# 3.5
* WPF Smart Client
* WPF Toolkit
* Microsoft Report Viewer

# Sicherheit

## Schutzmassnahmen

Es sind folgende Schutzmassnahmen gegen das „Hacken“ der Anwendung vorgesehen:

* **Gegen URL-Hacking**: In der URL werden keine Parameter mitgeschickt. Die URL selbst dient nur dazu, um am Applikationsserver eine konkrete Aktion eines Controllers auszulösen. Parameter werden rein im Message Body übertragen.
* **Gegen SQL Injection**: Die konsequente Verwendung eines O/R-Mappers macht SQL-Injection unmöglich.

## Benutzer- und Rechteverwaltung

### Authentifizierung

Die Authentifizierung erfolgt mittels ASP.NET Identity. Benutzer können sich mittels persönlichem Usernamen und Passwort einloggen.

#### Voraussetzungen für Benutzer

Damit sich Benturzer in EMSG-Master einloggen können, müssen sie mittels dem TestClient (Administrationsfunktion) angelegt werden.

### Autorisierung und Datensicherheit

Die Autorisierung der User wird ebenfalls mittels ASP.NET Identity durchgeführt. <https://www.asp.net/identity>

#### Rollen

Tabelle 10 zeigt eine Übersicht über die Zuordnung von Rollen zu Anwendungsfällen.

|  | **Mandantenweit**  **(Rollen auf Mandanten)** | | | | **Applikationsweit**  **(Applikationsweite Rollen)** | |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Data-Manager** | **Data-Reader** | **Benutzer-administrator** | **Benchmark-teilnehmer** | **Applikations-administrator** | **Applikations-supporter** |
| **N1-N5** | Ja |  |  |  |  |  |
| **Z1-Z6** | Ja |  |  |  |  |  |
| **K1-K2** | Ja |  |  |  |  |  |
| **R1-R3** | Ja |  |  |  |  |  |
| **F1** | Ja |  |  |  |  |  |
| **W1-W5** |  | Ja |  |  |  |  |
| **B1** |  |  |  | Ja |  |  |
| **A2-A5** |  |  | Ja |  |  |  |
| **A6-A9** |  |  |  |  | Ja |  |
| **A10** |  |  |  |  |  | Ja |
| **A11** | Ja | Ja | Ja | Ja | Ja | Ja |
| **A12** |  |  |  |  | Ja |  |
| **A13-A14** |  |  | Ja |  |  |  |

Tabelle 10: Zuordnung Rollen zu Anwendungsfällen

Es wird zwischen zwei Arten von Rollenzuordnungen unterschieden. Rollen, die ein Benutzer auf die Applikation EMSG hat (also Applikationsweit) und Rollen, die ein Benutzer für einen Mandanten hat (Mandantenweit).

#### Mandantenfähigkeit

Für den Mandanten in EMSG besteht ein Mapping zwischen (Gemeinde-Nr. (BFS) Attribut AstraMandatorName) und dem BS (OwnerId). Die Modellierung des Mandanten im Datenmodell ist in 4.4.4.2 dargestellt.

Ein Mandant in EMSG wird beim Einstieg in die Applikation gemäss des Mappings nach ASP.NET Identity identifiziert. Für den Bezug der Achsen aus dem BS wird gemäss des Mappings die OwnerId herangezogen.

Innerhalb von EMSG im Datenmodell wird die Zugehörigkeit der Fachdaten zu einem Mandanten mittels der Id (nur in EMSG-Master bekannt) des Mandanten realisiert.

Ein Strasseneigentümer in EMSG kann Gemeinde, Privat, Kanton oder Korporation sein. Der Strasseneigentümer wird auf Strassenabschnitte definiert und hat keinen Bezug zu ASP.NET Identity oder dem BS.

##### Sichtbarkeit EMSG – GIS Visualisierung

Der GIS-Modus am EMSG-Master visualisiert auf dem Kartenelement immer nur die dem Mandanten zugehörigen Daten. Diese Einschränkung wird technisch durch die Filterung der Daten bei der Abfrage umgesetzt. Aus Performancesicht ist dieser Ansatz nicht kritisch, da nur mehr die ensprechenden Daten übertragen werden.

Um sicherzustellen, dass autorisierte Clients nur die erlaubten Daten angezeigt bekommen, wird in der EMSG-Master Geschäftslogik ein Security-Proxy implementiert. Dieser hat die Aufgabe, die jeweilige Session-ID auf die jeweilig zugehörige Mandanten-ID zu übersetzen. Die Abfrage der Daten erfolgt somit direkt am System.

#### Mandantenfähigkeit GIS-Daten

Im DBMS sind alle Datensätze einer Entität in der selben Tabelle gespeichert. Jeder GIS Datensatz ist über das Attribut MandantID einem Mandanten eindeutig zugeordnet.

* **Editierbarkeit**: Anwender können alle Datensätze manipulieren, wenn diese Ihrem Mandanten zugeordnet sind. Datensätze von anderen Mandanten können nicht editiert, angelegt oder gelöscht werden.
* **Sichtbarkeit**: Dem Anwender werden prinzipiell nur jene Datensätze in der Karte und tabellarischer Form angezeigt, wenn diese die entsprechende MandantID besitzen. Ausnahme: Achsen sind für alle Mandanten in der Karte sichtbar. EMSG-Objekte (Strassenabschnitte, Zustandsabschnitte etc.) können aber nur auf dem Mandanten zugewiesenen Achssegmenten erzeugt werden.

#### Einschränkung der Sichtbarkeit – GeoServer

Der GeoServer stellt Karteninhalte primär in Form von WMS-Services zur Verfügung. Um sicherzustellen, dass nur für den Mandanten sichtbare Daten angezeigt werden, steht für jeden Mandanten ein eigener WMS Layer (ein WMS-Service besteht aus n optional gruppierten WMS Layern, welche beim Request angegeben werden) zur Verfügung, der die entsprechende Einschränkung in Form von „Definition Querys“ abbildet.

Um zu gewährleisten, dass ein Mandant nur die ihm zugeordneten Layer laden und anzeigen kann, wird zwischen dem EMSG-Master WebClient und EMSG GeoServer ein Security Proxy (schemenhaft illustriert in Abbildung 64) zwischengeschalten, welcher bei jedem Request prüft, ob der Mandant die angeforderten Layer anzeigen, bzw. anfragen darf. Der Security-Proxy ist in der EMSG Geschäftslogik angesiedelt.

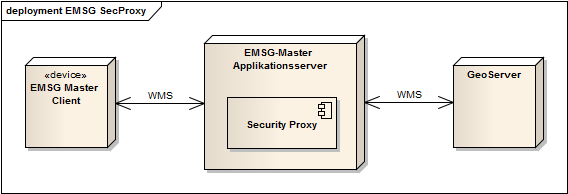


Abbildung 64: Schematische Abbildung des SecurityProxies für die Einschränkung der Sichtbarkeit von WMS Services

Anmerkung: Diese Vorgangsweise erlaubt es prinzipiell in der Zukunft unterschiedliche Styles je nach Mandant zu definieren.

#### Support

Ein Benutzer mit der Rolle „Applikationssupporter“ kann im Zusammenhang mit Supportanfragen von Benutzern die Zugriffsrechte des entsprechenden Organisationsanwenders (d.h. alle Rollen ausser Applikationsadministrator) für die Dauer der Supportanfrage übernehmen (siehe EMSG Systemanforderungen [7], Anwendungsfall A10).

Der Applikationssupporter nimmt die Rollen anderer Benutzer an, nicht deren Identität. Deshalb soll, falls gewisse Objektmutationen protokolliert werden, im Protokoll der Name des Applikationssupporters erscheinen, und nicht der des Benutzers.

#### EMSG-Mobile

Da die mobilen Endgeräte offline betrieben werden, steht keine technische Möglichkeit zur Verfügung, um eine Autorisierung gegen ASP.NET Indentiy durchzuführen.

Jeder, der physischen Zugriff auf ein mobiles Endgerät hat, kann in EMSG-Master exportierte Daten (z.B. von einem USB-Stick) in EMSG-Mobile importieren.

#### Einstieg in EMSG-Master

Beim Einstieg in EMSG-Master werden im ersten Schritt die Rollen des Benutzers aus ASP.NET Indentiy gelesen. An Hand der Rollen wird bestimmt ob der Benutzer mit applikationsweiten Rollen oder mit den Rollen seines Mandanten angemeldet wird.

Falls der Benutzer nur Rollen auf einen Mandanten besitzt so wird er für diesen Mandanten im System angemeldet. Hat der Benutzer Rollen auf mehrere Mandanten so wird er für jenen Mandanten im System angemeldet für welchen er zuletzt auf seinem Rechner angemeldet war (Auswahl wird in einem Cookie am Rechner gespeichert). Meldet er sich zum ersten Mal an so entscheidet EMSG-Master für welchen seiner Mandanten er angemeldet wird.

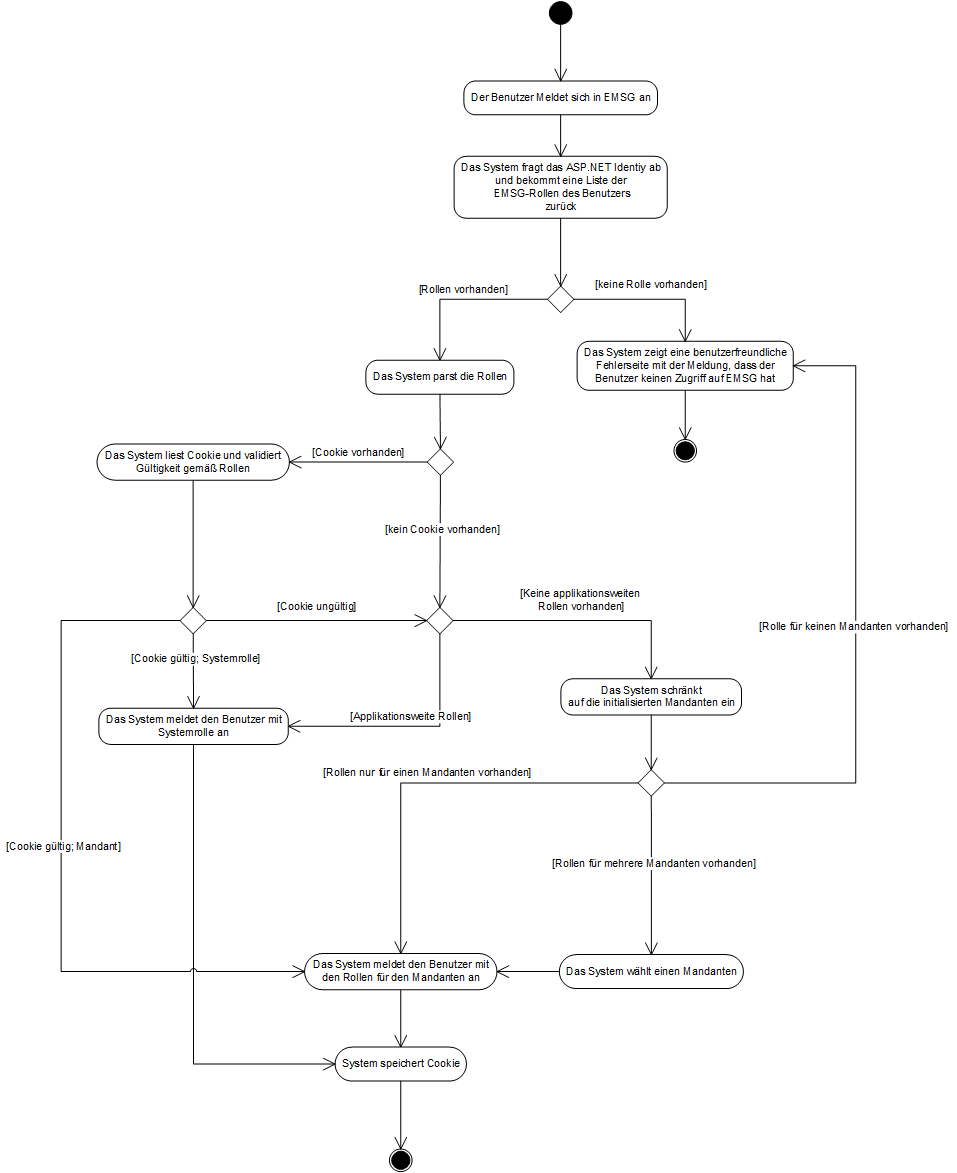
Hat ein Benutzer nur applikationsweite Rollen so wird er immer automatisch mit diesen Rollen angemeldet. Unabhängig davon ob er auch Rollen auf Mandanten besitzt.

Abbildung 65 veranschaulicht graphisch den Einstieg in EMSG-Master.

Abbildung 65: Einstieg in EMSG-Master

#### Mandantenauswahl

Falls der Benutzer für mehrere Mandanten Rollen besitzt so hat er innerhalb der Applikation die Möglichkeit den aktuellen Mandanten zu wechseln. Ändert der Benutzer den aktuellen Mandanten so wird er sofort mit diesem abgemeldet und mit dem neuen Mandanten im System angemeldet.

Hat der Benutzer neben den Rollen auf Mandanten auch applikationsweite Rollen so kann er entweder die Rollen eines Mandanten oder applikationsweite Rollen einnehmen. Er kann aber nie Rollen eines Mandanten kombiniert mit applikationsweiten Rollen einnehmen. 

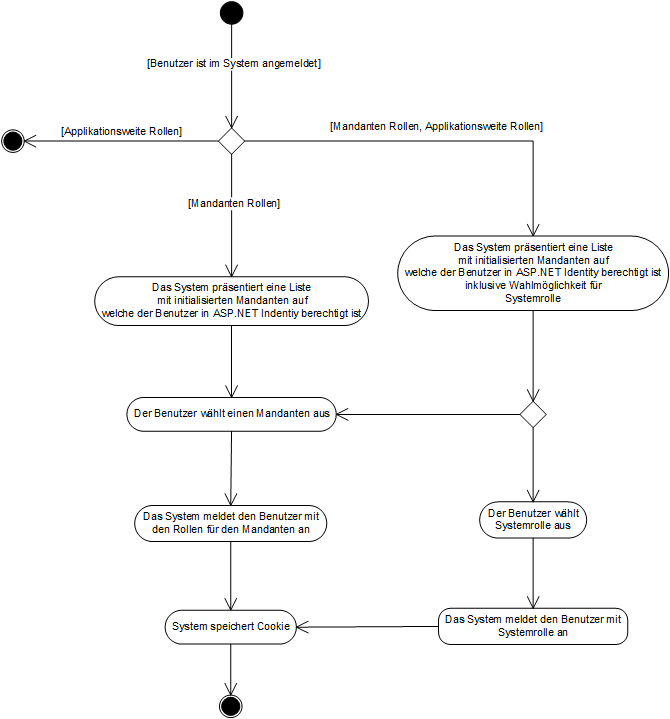


Abbildung 66: Mandantenauswahl

# Wartung

Diese Kapitel richtet sich an Entwickler welche Anpassungen im Code vornehmen wollen. Es werden folgenden Bereiche beschrieben:

* Quellcode (Aufbau und Gliederung)
* Wichtige Konfigurationsmöglichkeiten
* Einrichten der Entwicklungsmaschine
* Ausführung Unit-/Integration Tests Einleitung
* Developer Client
* Physisches Datenmodell

## Quellcode

Der Quellcode ist zwecks Übersichtlichkeit, Wartbarkeit und Wiederverwendung in mehrere Solutions unterteilt. Die Einteilung wurde in folgende Solutions vorgenommen:

* Common (Projekte welche in Master und Mobile verwendet werden)
* Master (beinhaltet alle relevanten Projekte für EMSG-Master)
* Mobile (beinhaltet alle relevanten Projekte für EMSG-Mobile)

Die externen Bibliotheken befinden sich in Verzeichnissen Libraries (manuelles Management) bzw. Packages (Manangement durch NuGet[[38]](#footnote-39)) der jeweiligen Solution.

### Common

Im Folgenden werden die C# Projekte der Solution Common beschrieben.

#### ASTRA.EMSG.Common

Enthalten sind hier alle Klassen welche in Master sowohl Mobile für den Datenaustausch Verwendung finden.

### Master

Im Folgenden werden die C# Projekte der Solution Master beschrieben.

#### ASTRA.EMSG.AxisImportService

Windows Service für den Import der Achsen aus Interlis2 Dateien. Dieses Projekt beinhaltet die Infrastruktur für ein Windows Service. Das Windows Service verwendet die Geschäftslogik von 9.1.2.2.

#### ASTRA.EMSG.Business

In diesem Projekt ist die Geschäftslogik implementiert. Es handelt sich dabei um folgende Bereiche:

* Definition von Entitäten plus Mapping auf das Datenbankschema
* Definition der Reports (rdlc)
* Abbildung der Models
* Transformation von Entitäten <-> Models mittels Object Mapper
* Implementierung der Entityservices aller Anwendungsfälle und Bereitstellung von CRUD Operation
* Lokalisierung
* Validierung von Models
* Importieren von Achsen
* Securityprüfungen
* Logging

#### ASTRA.EMSG.Common.Master

Verschiedene Hilfsklassen für die Geschäftslogik.

#### ASTRA.EMSG.IntegrationTests

Beinhaltet alle Integrationstests welche mit SpecFlow erstellt wurden. Als Test-Framework wird NUnit verwendet. Für die Testausführung wird eine SQL-Lite Datenbank verwendet.

Zur Ausführung werden keine weiteren Fremdsysteme verwendet.

#### ASTRA.EMSG.TestClient

Developer Client welcher während der Entwicklung zur Unterstützung entwickelt und eingesetzt wurde. Nähere Informationen dazu siehe 9.5.

#### ASTRA.EMSG.Tests

Beinhaltet alle Unit Tests. Als Test-Framework wird NUnit verwendet. Alle Testfälle laufen in Memory.

Zur Ausführung werden keine weiteren Fremdsysteme verwendet.

#### ASTRA.EMSG.Tests.Common

Stellt gemeinsam genutzte Infrastruktur für 9.1.2.4, **Error! Reference source not found.**, 9.1.2.5 und 9.1.2.6 dar.

#### ASTRA.EMSG.Web

Implementierung der Masken/Screens mittels Views in MVC für EMSG-Master. Die Unterteilung erfolgt technisch basierend auf fachlicher Ebene in folgende 4 Areas in MVC:

* Administration
* Auswertungen
* Benchmarking
* Common
* NetzverwaltungGIS
* NetzverwaltungSummarisch
* NetzverwaltungTabellarisch

In jeder Area sind die entsprechenden Controller und Views der Anwendungsfälle implementiert.

Die verwendeten Models sind in 9.1.2.2 ausgelagert. Das Projekt verwendet die Geschäftslogik aus 9.1.2.2. Aufrufe der Action werden an die Geschäftslogik weiterdelegiert.

### Mobile

Im Folgenden werden die C# Projekte der Solution Mobile beschrieben.

#### ASTRA.EMSG.Map

Beinhaltet das Karten-Control primär bestehend aus einem WPF WebBrowser-Control.

#### ASTRA.EMSG.Localization

Klassen für die Lokalisierung im WPF Client.

#### ASTRA.EMSG-Mobile

Beinhaltet alle Views, ViewModels, Models und Services des EMSG-Mobile. Hier ist die eigentliche Businesslogik und die Präsentationsschicht abgebildet.

#### ASTRA.EMSG-Mobile.Installer

Installationsprogramm für EMSG-Mobile.

#### ASTRA.EMSG-Mobile.InstallerCreator

Erweitert Installationsprogramm um EMSG-Mobile.

## Wichtige Konfigurationsmöglichkeiten

Alle konfigurationsmöglichkeiten werden im EMGS Konfigurationshandbuch web.config näher erläutert.

## Einrichten der Entwicklungsmaschine

Dieser Abschnitt beschreibt welche Schritte für die Konfiguration der Entwicklungsmaschine durchzuführen sind.

### Entwicklungsumgebung

* Installation Visual Studio 2010 Professional (oder höher) SP1
* Install Team Foundation Server Tools[[39]](#footnote-40) (optional)
* Installation MVC3[[40]](#footnote-41)
* Tools -> Extension Manager 🡪 Update NuGet Package Manager (mindestens Version 1.4)
* Tools -> Extension Manager 🡪 Install Razor Generator

### IIS

* IISExpress[[41]](#footnote-42) 7.0 oder
* Lokaler IIS 7.0

### Frameworks

* Microsoft .NET Framework 4.0 (Bestandteil von Microsoft Visual Studio 2010, siehe 9.3.1)

Microsoft Visual C++ 2010 Redistributable Package

### Tools

Folgende Tools wurden während der Entwicklung eingesetzt und werden für die Wartung empfohlen:

* SQLServer Management Studio
* Firebug[[42]](#footnote-43)

## Unit-/Integration Tests

Bei den automatisierten Tests wird das Nunit Test-Framework verwendet. Zur Ausführung der Tests werden folgende Testrunner empfohlen:

* Resharper[[43]](#footnote-44) (kostenpflichtig)
* TestDrivenDotNet[[44]](#footnote-45) (kostenpflichtig)
* NUnit[[45]](#footnote-46) (keine Lizenz notwendig)

## Developer Client

Hierbei handelt es sich um ein Hilfswerkzeug für die Entwicklung. **Der Developer Client ist nicht für den produktiven Einsatz vorgesehen.**

Abbildung 67 zeigt die Startseite des Developer Client.

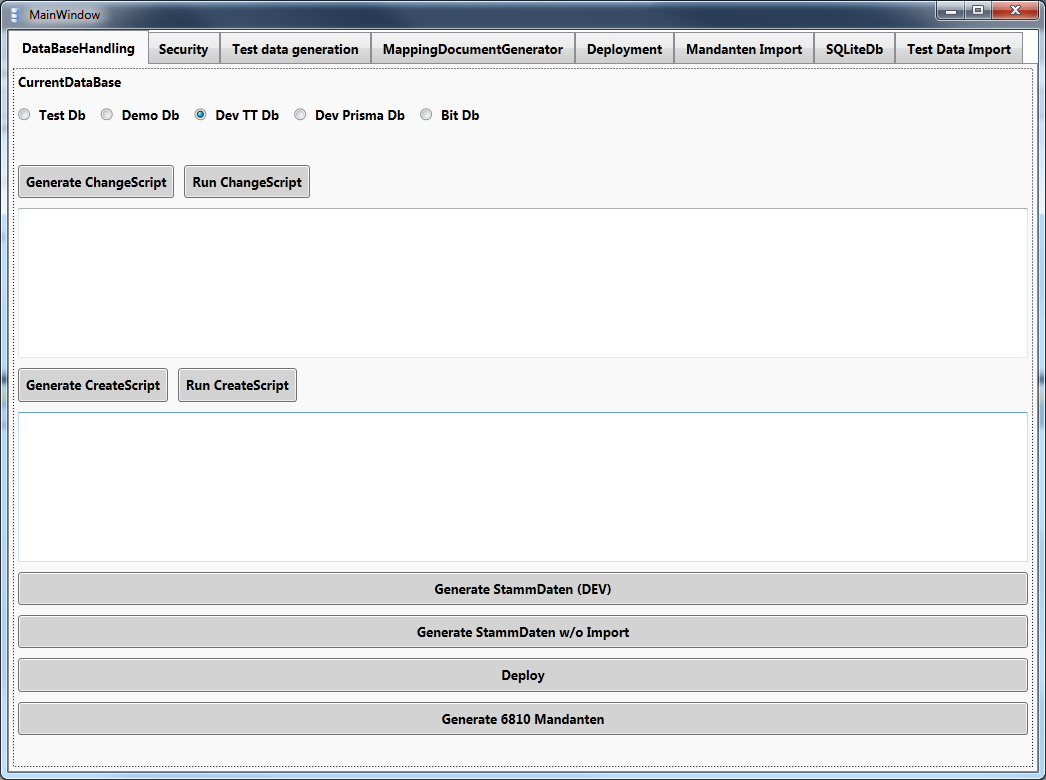


Abbildung 67: Developer Client

Die verschiedenen Funktionen sind auf folgende Tabs verteilt:

* DatabaseHandling
* Security
* Test data generation
* MappingDocumentGenerator
* Deployment
* Mandanten Import
* SQLiteDb
* Test Data Import

Im nachfolgenden werden die wichtigsten Funktionen erklärt.

### Datenbankschema

Das Erzeugen des Datenbankschema erfolgt über die Funktion „Generate Createscripts“. Dabei wird das vollständige Datenbankschema von EMSG-Master generiert. Das Datenbankschema wird dabei in der Messagebox ausgegeben und kann über die Zwischenablage kopiert und anschliessend weiterverwendet werden.

Abbildung 68 zeigt die Ausgabe in der Messagebox.

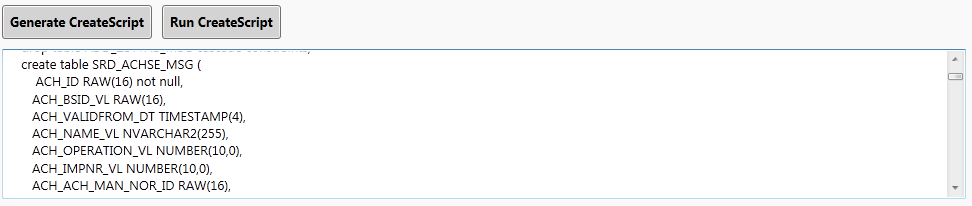


Abbildung 68: Erzeugen des Datenbankschema

Mittels der Funktion „Run CreateScript“ wird das Datenbankschema in MSSQL erstellt. Dazu ist die entsprechende MSSQL Instance zu wählen. Es sind 5 verschiedenen Datenbankverbindungen vorkonfiguriert.

Abbildung 69 zeigt die Auswahl der Datenbank Instance.

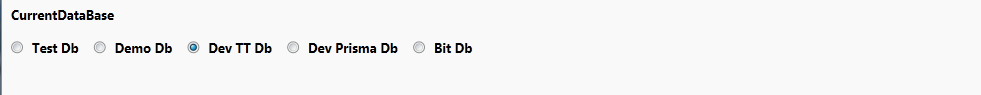


Abbildung 69: Auswahl Datenbank Instance

### Erzeugen von Stammdaten

Mit dieser Funktion wird eine kleine Anzahl von Mandanten in MSSQL erzeugt.

Achsen werden dabei optional in MSSQL eingespielt, siehe Abbildung 70 und Abbildung 71.



Abbildung 70: Erzeugen von Stammdaten mit Achsen



Abbildung 71: Erzeugen von Stammdaten ohne Achsen

### Bearbeiten der Berechtigungen

Abbildung 72 zeigt das Bearbeiten des Benutzers ue02464.

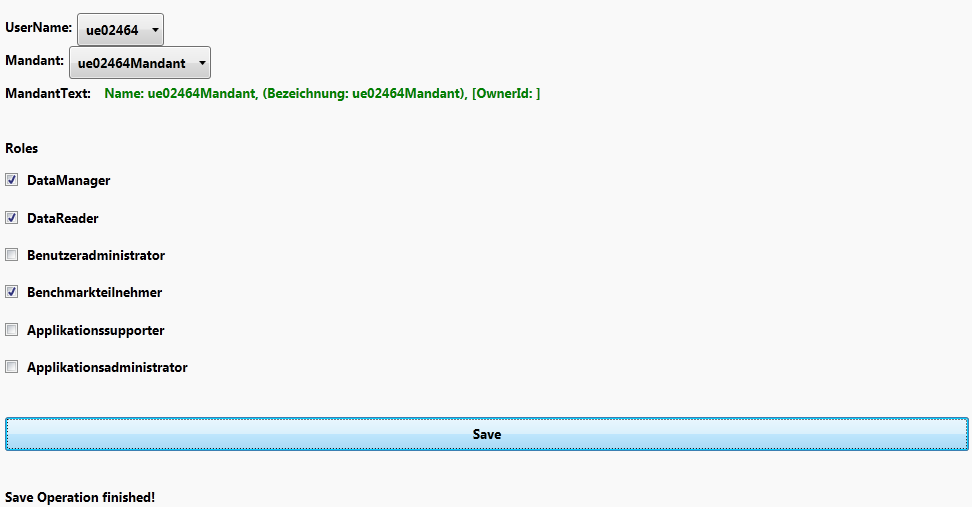


Abbildung 72: Bearbeiten von Berechtigungen

### Erzeugen von Testdaten

Mit dieser Funktion können die Fachdaten erzeugt und gelöscht werden bzw. ein Achsenupdate durchgeführt werden. Die Funktion bezieht sich immer auf den ausgewählten Mandanten.

Abbildung 73 zeigt die Funktion.

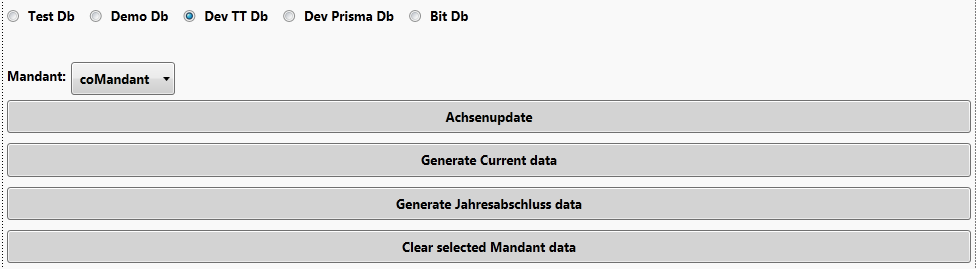


Abbildung 73: Erzeugen von Testdaten

### Deployment Package

Dadurch werden die Binaries für EMSG-Master, EMSG-Mobile und dem AchsenImportService für ein Deployment zur Verfügung gestellt.

Abbildung 74 zeigt die Erstellung des Deployment Package.



Abbildung 74: Erstellen des Deployment Package

Das Deployment Package wird im Binary Verzeichnis des Developer Client erstellt und befindet sich in „DeploymentDropLocation“.

### SQLiteDb

Alternative zu MSSQL für Integrationstests.

### Test Data Import

Import von Testdaten aus Excel. Fachdaten und Jahresabschlüsse aller Mandanten werden dabei gelöscht.

## Physisches Datenmodell

### Mapping Quellcode zu EMSG Naming Conventions.

|  |  |  |
| --- | --- | --- |
| **Achse** | **=** | **SRD\_ACHSE\_MSG** |
| Id | : | ACH\_ID |
| AchsenSegmente | : |  |
| BsId | : | ACH\_BSID\_VL |
| VersionValidFrom | : | ACH\_VALIDFROM\_DT |
| Name | : | ACH\_NAME\_VL |
| Operation | : | ACH\_OPERATION\_VL |
| ImpNr | : | ACH\_IMPNR\_VL |
| Mandant | : | ACH\_ACH\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | ACH\_ACH\_EPR\_NOR\_ID |
| CopiedFrom | : | ACH\_ACH\_ACH\_NOR\_ID |
|  | | |
|
| **AchsenImportLog** | **=** | **ADD\_ACHSIMPLOG\_MSG** |
| Id | : | AIL\_ID |
| ImpNr | : | AIL\_IMPNR\_VL |
| Path | : | AIL\_PATH\_VL |
| Progress | : | AIL\_PROGRESS\_VL |
| SenderTimestamp | : | AIL\_SENDERTIMES\_DT |
| Timestamp | : | AIL\_TIMESTAMP\_DT |
| AchsInserts | : | AIL\_ACHSINSERTS\_VL |
| AchsUpdates | : | AIL\_ACHSUPDATES\_VL |
| AchsDeletes | : | AIL\_ACHSDELETES\_VL |
| SegmInserts | : | AIL\_SEGMINSERTS\_VL |
| SegmUpdates | : | AIL\_SEGMUPDATES\_VL |
| SegmDeletes | : | AIL\_SEGMDELETES\_VL |
| SektInserts | : | AIL\_SEKTINSERTS\_VL |
| SektUpdates | : | AIL\_SEKTUPDATES\_VL |
| SektDeletes | : | AIL\_SEKTDELETES\_VL |
|  | | |
|
| **AchsenLock** | **=** | **ADD\_ACHSLOCK\_MSG** |
| Id | : | ALK\_ID |
| IsLocked | : | ALK\_ISLOCKED\_VL |
| LockStart | : | ALK\_LOCKSTART\_DT |
| LockEnd | : | ALK\_LOCKEND\_VL |
| LockType | : | ALK\_LOCKTYPE\_VL |
| Mandant | : | ALK\_ALK\_MAN\_NOR\_ID |
|  | | |
|
| **AchsenReferenz** | **=** | **ADD\_ACHSREF\_MSG** |
| Id | : | ACR\_ID |
| Strassenname | : | ACR\_STRASSENNAM\_VL |
| Version | : | ACR\_VERSION\_VL |
| VonRBBS | : | ACR\_VONRBBS\_VL |
| NachRBBS | : | ACR\_NACHRBBS\_VL |
| Shape | : | ACR\_SHAPE\_VL |
| CopiedFrom | : | ACR\_ACR\_ACR\_NOR\_ID |
| AchsenSegment | : | ACR\_ACR\_ACS\_NOR\_ID |
| ReferenzGruppe | : | ACR\_ACR\_RFG\_NOR\_ID |
|  | | |
|
| **AchsenSegment** | **=** | **SRD\_ACHSSEG\_MSG** |
| Id | : | ACS\_ID |
| Sektoren | : |  |
| AchsenReferenzen | : |  |
| BsId | : | ACS\_BSID\_VL |
| Operation | : | ACS\_OPERATION\_VL |
| Name | : | ACS\_NAME\_VL |
| Sequence | : | ACS\_SEQUENCE\_VL |
| ImpNr | : | ACS\_IMPNR\_VL |
| Shape | : | ACS\_SHAPE\_VL |
| Shape4d | : | ACS\_SHAPE4D\_VL |
| Version | : | ACS\_VERSION\_VL |
| AchsenId | : | ACS\_ACHSENID\_VL |
| Achse | : | ACS\_ACS\_ACH\_NOR\_ID |
| ErfassungsPeriod | : | ACS\_ACS\_EPR\_NOR\_ID |
| Mandant | : | ACS\_ACS\_MAN\_NOR\_ID |
| CopiedFrom | : | ACS\_ACS\_ACS\_NOR\_ID |
|  | | |
|
| **AchsenUpdateConflict** | **=** | **ADD\_ACHSUPDCON\_MSG** |
| Id | : | AUC\_ID |
| Name | : | AUC\_NAME\_VL |
| ConflictType | : | AUC\_CONFLICTTYP\_VL |
| ItemType | : | AUC\_ITEMTYPE\_VL |
| ItemId | : | AUC\_ITEMID\_VL |
| SegmentId | : | AUC\_SEGMENTID\_VL |
| Shape | : | AUC\_SHAPE\_VL |
| Mandant | : | AUC\_AUC\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | AUC\_AUC\_EPR\_NOR\_ID |
|  | | |
|
| **AchsenUpdateLog** | **=** | **ADD\_ACHSUPDLOG\_MSG** |
| Id | : | AUL\_ID |
| ImpNr | : | AUL\_IMPNR\_VL |
| Statistics | : | AUL\_STATISTICS\_VL |
| Timestamp | : | AUL\_TIMESTAMP\_DT |
| AchsInserts | : | AUL\_ACHSINSERTS\_VL |
| AchsUpdates | : | AUL\_ACHSUPDATES\_VL |
| AchsDeletes | : | AUL\_ACHSDELETES\_VL |
| SegmInserts | : | AUL\_SEGMINSERTS\_VL |
| SegmUpdates | : | AUL\_SEGMUPDATES\_VL |
| SegmDeletes | : | AUL\_SEGMDELETES\_VL |
| SektInserts | : | AUL\_SEKTINSERTS\_VL |
| SektUpdates | : | AUL\_SEKTUPDATES\_VL |
| SektDeletes | : | AUL\_SEKTDELETES\_VL |
| UpdatedReferences | : | AUL\_UPDATEDREFE\_VL |
| DeletedReferences | : | AUL\_DELETEDREFE\_VL |
| UpdatedStrassenabschnitts | : | AUL\_UPDATEDSTRA\_VL |
| DeletedStrassenabschnitts | : | AUL\_DELETEDSTRA\_VL |
| UpdatedZustandsabschnitts | : | AUL\_UPDATEDZUST\_VL |
| DeletedZustandsabschnitts | : | AUL\_DELETEDZUST\_VL |
| UpdatedKoordinierteMassnahmen | : | AUL\_UPDATEDKOOR\_VL |
| DeletedKoordinierteMassnahmen | : | AUL\_DELETEDKOOR\_VL |
| UpdatedMassnahmenvorschlagTeilsysteme | : | AUL\_UPDATEDMASS\_VL |
| DeletedMassnahmenvorschlagTeilsysteme | : | AUL\_DELETEDMASS\_VL |
| Mandant | : | AUL\_AUL\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | AUL\_AUL\_EPR\_NOR\_ID |
|  | | |
|
| **Belastungskategorie** | **=** | **VAT\_BELKAT\_MSG** |
| Id | : | BLK\_ID |
| AllowedBelagList | : |  |
| Typ | : | BLK\_TYP\_VL |
| Reihenfolge | : | BLK\_REIHENFOLGE\_VL |
| DefaultBreiteFahrbahn | : | BLK\_DEFBRFB\_VL |
| DefaultBreiteTrottoirRechts | : | BLK\_DEFBRTRR\_VL |
| DefaultBreiteTrottoirLinks | : | BLK\_DEFBRTRL\_VL |
| ColorCode | : | BLK\_FARBCODE\_VL |
|  | | |
|
| **BenchmarkingDataDetail** | **=** | **ADD\_BEDATADET\_MSG** |
| Id | : | BDD\_ID |
| FahrbahnflaecheAnteil | : | BDD\_FBFLANTEIL\_NR |
| Zustandsindex | : | BDD\_ZUSTANDSIND\_VL |
| RealisierteMassnahmenProWiederbeschaffungswertNetz | : | BDD\_REMAPWBW\_NR |
| Belastungskategorie | : | BDD\_BDD\_BLK\_NOR\_ID |
| BenchmarkingData | : | BDD\_BDD\_BDT\_NOR\_ID |
|  | | |
|
| **BenchmarkingData** | **=** | **ADD\_BENCHDATA\_MSG** |
| Id | : | BDT\_ID |
| BenchmarkingDataDetails | : |  |
| FahrbahnflaecheProEinwohner | : | BDT\_FAFLPEIN\_NR |
| FahrbahnflaecheProSiedlungsflaeche | : | BDT\_FAFLPSIE\_NR |
| GesamtlaengeDesStrassennetzesProEinwohner | : | BDT\_GESTRPEIN\_NR |
| GesamtlaengeDesStrassennetzesProSiedlungsflaeche | : | BDT\_GESTRPSIE\_NR |
| GesamtstrassenflaecheProEinwohner | : | BDT\_GESTFLPEIN\_NR |
| GesamtstrassenflaecheProSiedlungsflaeche | : | BDT\_GESTFLPSIE\_NR |
| WertverlustProEinwohner | : | BDT\_WVLPEIN\_NR |
| WertverlustProFahrbahn | : | BDT\_WVLPFB\_NR |
| WiederbeschaffungswertProEinwohner | : | BDT\_WBWPEIN\_NR |
| WiederbeschaffungswertProFahrbahn | : | BDT\_WBWPFB\_NR |
| ZustandsindexNetz | : | BDT\_ZUSTANDNETZ\_VL |
| MittleresAlterDerZustandsaufnahmenNetz | : | BDT\_MITALTZUSNE\_VL |
| RealisierteMassnahmenProEinwohner | : | BDT\_REMAPEIN\_NR |
| RealisierteMassnahmenProFahrbahn | : | BDT\_REMAPFB\_NR |
| RealisierteMassnahmenProWertverlustNetz | : | BDT\_REMAWV\_NR |
| RealisierteMassnahmenProWiederbeschaffungswertNetz | : | BDT\_REMAWBW\_NR |
| ErfassungsPeriod | : | BDT\_BDT\_EPR\_NOR\_ID |
| Mandant | : | BDT\_BDT\_MAN\_NOR\_ID |
|  | | |
|
| **BenchmarkingGruppenConfiguration** | **=** | **ADD\_BENCHGRCFG\_MSG** |
| Id | : | BGC\_ID |
| EigenschaftTyp | : | BGC\_EIGENSCHAFT\_VL |
| Grenzwert | : | BGC\_GRENZWERT\_NR |
|  | | |
|
| **CheckOutsGIS** | **=** | **ADD\_CHECKOUT\_MSG** |
| Id | : | COG\_ID |
| CheckInDatum | : | COG\_CHECKINDATU\_VL |
| CheckOutDatum | : | COG\_CHECKOUTDAT\_DT |
| CheckedOutUntil | : | COG\_CHECKEDOUTU\_DT |
| InspectionBy | : | COG\_INSPECTIONB\_VL |
| Description | : | COG\_DESCRIPTION\_VL |
| Comments | : | COG\_COMMENTS\_VL |
| Mandant | : | COG\_COG\_MAN\_NOR\_ID |
| InspektionsRouteGIS | : | COG\_COG\_IRG\_NOR\_ID |
|  | | |
|
| **EreignisLog** | **=** | **ADD\_EREIGNISLOG\_MSG** |
| Id | : | ERL\_ID |
| Benutzer | : | ERL\_BENUTZER\_VL |
| Zeit | : | ERL\_ZEIT\_DT |
| EreignisTyp | : | ERL\_EREIGNISTYP\_VL |
| EreignisData | : | ERL\_EREIGNISDAT\_VL |
| MandantName | : | ERL\_MANDANTNAME\_VL |
|  | | |
|
| **ErfassungsPeriod** | **=** | **ADD\_ERFPERIODE\_MSG** |
| Id | : | EPR\_ID |
| Name | : | EPR\_NAME\_VL |
| NetzErfassungsmodus | : | EPR\_NETZMODUS\_VL |
| IsClosed | : | EPR\_ISTABGESCHL\_VL |
| Erfassungsjahr | : | EPR\_ERFJAHR\_DT |
| Mandant | : | EPR\_EPR\_MAN\_NOR\_ID |
|  | | |
|
| **GemeindeKatalog** | **=** | **VAT\_GEMKAT\_MSG** |
| Id | : | GEK\_ID |
| Typ | : | GEK\_TYP\_VL |
|  | | |
|
| **GlobalMassnahmenvorschlagKatalog** | **=** | **VAT\_GMASSVOR\_MSG** |
| Id | : | GMK\_ID |
| Typ | : | GMK\_TYP\_VL |
| DefaultKosten | : | GMK\_DEFKOSTEN\_NR |
| KatalogTyp | : | GMK\_KATALOGTYP\_VL |
| Belastungskategorie | : | GMK\_GMK\_BLK\_NOR\_ID |
|  | | |
|
| **GlobalRealisierteMassnahmenvorschlagKatalog** | **=** | **VAT\_GRMASSVOR\_MSG** |
| Id | : | GRK\_ID |
| Typ | : | GRK\_TYP\_VL |
|  | | |
|
| **GlobalWiederbeschaffungswertKatalog** | **=** | **VAT\_GWBBKAT\_MSG** |
| Id | : | GWK\_ID |
| GesamtflaecheFahrbahn | : | GWK\_FLAEGESFB\_NR |
| FlaecheFahrbahn | : | GWK\_FLAECHEFB\_NR |
| FlaecheTrottoir | : | GWK\_FLAECHETR\_NR |
| Gesamtlaenge | : | GWK\_GESAMTLAENG\_NR |
| GesamtflaecheFahrbahnTrottoir | : | GWK\_FALEGESFBTR\_NR |
| AlterungsbeiwertI | : | GWK\_ALTERUNGI\_NR |
| AlterungsbeiwertII | : | GWK\_ALTERUNGII\_NR |
| Belastungskategorie | : | GWK\_GWK\_BLK\_NOR\_ID |
|  | | |
|
| **InspektionsRouteGIS** | **=** | **ADD\_INSPEKROUTE\_MSG** |
| Id | : | IRG\_ID |
| CheckOutsGISList | : |  |
| InspektionsRtStrAbschnitteList | : |  |
| StatusverlaufList | : |  |
| Shape | : | IRG\_SHAPE\_VL |
| Bezeichnung | : | IRG\_BEZEICHNUNG\_VL |
| Bemerkungen | : | IRG\_BEMERKUNGEN\_VL |
| Beschreibung | : | IRG\_BESCHREIBUN\_VL |
| InInspektionBei | : | IRG\_ININSPBEI\_VL |
| InInspektionBis | : | IRG\_ININSPBIS\_VL |
| LegendNumber | : | IRG\_LEGENDNUMBE\_VL |
| Mandant | : | IRG\_IRG\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | IRG\_IRG\_EPR\_NOR\_ID |
|  | | |
|
| **InspektionsRouteStatusverlauf** | **=** | **ADD\_INSPEKSTATU\_MSG** |
| Id | : | IRV\_ID |
| Datum | : | IRV\_DATUM\_DT |
| Status | : | IRV\_STATUS\_VL |
| InspektionsRouteGIS | : | IRV\_IRV\_IRG\_NOR\_ID |
|  | | |
|
| **InspektionsRtStrAbschnitte** | **=** | **ADD\_INSPEKSTRA\_MSG** |
| Id | : | IRS\_ID |
| Reihenfolge | : | IRS\_REIHENFOLGE\_VL |
| StrassenabschnittGIS | : | IRS\_IRS\_STG\_NOR\_ID |
| InspektionsRouteGIS | : | IRS\_IRS\_IRG\_NOR\_ID |
|  | | |
|
| **KenngroessenFruehererJahreDetail** | **=** | **ADD\_KENGRFJDET\_MSG** |
| Id | : | KFD\_ID |
| MittlererZustand | : | KFD\_MITZST\_VL |
| Fahrbahnlaenge | : | KFD\_FBLAENGE\_NR |
| Fahrbahnflaeche | : | KFD\_FBFLAECHE\_VL |
| Belastungskategorie | : | KFD\_KFD\_BLK\_NOR\_ID |
| KenngroessenFruehererJahre | : | KFD\_KFD\_KFJ\_NOR\_ID |
|  | | |
|
| **KenngroessenFruehererJahre** | **=** | **ADD\_KENGRFJ\_MSG** |
| Id | : | KFJ\_ID |
| KenngroesseFruehereJahrDetails | : |  |
| Jahr | : | KFJ\_JAHR\_VL |
| MittlerenJaehrlicherWerteverlust | : | KFJ\_MITTLERENJA\_NR |
| KostenFuerWerterhaltung | : | KFJ\_KOSTENFUERW\_NR |
| Mandant | : | KFJ\_KFJ\_MAN\_NOR\_ID |
|  | | |
|
| **KoordinierteMassnahmeGIS** | **=** | **ADD\_KOORMASSGIS\_MSG** |
| Id | : | KMG\_ID |
| BeteiligteSysteme | : |  |
| Projektname | : | KMG\_PROJEKTNAME\_VL |
| BezeichnungVon | : | KMG\_BEZVON\_VL |
| BezeichnungBis | : | KMG\_BEZBIS\_VL |
| Laenge | : | KMG\_LAENGE\_NR |
| BreiteFahrbahn | : | KMG\_BREITEFB\_NR |
| BreiteTrottoirLinks | : | KMG\_BREITETRL\_VL |
| BreiteTrottoirRechts | : | KMG\_BREITETRR\_VL |
| KostenGesamtprojekt | : | KMG\_KOSTGESAMT\_VL |
| KostenFahrbahn | : | KMG\_KOSTENFB\_VL |
| KostenTrottoirLinks | : | KMG\_KOSTENTRL\_VL |
| KostenTrottoirRechts | : | KMG\_KOSTENTRR\_VL |
| Beschreibung | : | KMG\_BESCHREIBUN\_VL |
| AusfuehrungsAnfang | : | KMG\_AUSFANF\_VL |
| AusfuehrungsEnde | : | KMG\_AUSFENDE\_VL |
| LeitendeOrganisation | : | KMG\_LEITENDEORG\_VL |
| Status | : | KMG\_STATUS\_VL |
| Shape | : | KMG\_SHAPE\_VL |
| MassnahmenbeschreibungFahrbahn | : | KMG\_KMG\_RMK\_NOR\_ID |
| ReferenzGruppe | : | KMG\_KMG\_RFG\_NOR\_ID |
| Mandant | : | KMG\_KMG\_MAN\_NOR\_ID |
|  | | |
|
| **KopieAchse** | **=** | **SRD\_KOPACHSE\_MSG** |
| Id | : | KAC\_ID |
| ImpNr | : | KAC\_IMPNR\_VL |
| Name | : | KAC\_NAME\_VL |
| Operation | : | KAC\_OPERATION\_VL |
| Owner | : | KAC\_OWNER\_VL |
| VersionValidFrom | : | KAC\_VALIDFROM\_DT |
|  | | |
|
| **KopieAchsenSegment** | **=** | **SRD\_KOPACHSSEG\_MSG** |
| Id | : | KSG\_ID |
| ImpNr | : | KSG\_IMPNR\_VL |
| Name | : | KSG\_NAME\_VL |
| Operation | : | KSG\_OPERATION\_VL |
| Sequence | : | KSG\_SEQUENCE\_VL |
| Shape | : | KSG\_SHAPE\_VL |
| AchsenId | : | KSG\_KSG\_KAC\_NOR\_ID |
|  | | |
|
| **KopieSektor** | **=** | **SRD\_KOPSEKTOR\_MSG** |
| Id | : | KSK\_ID |
| ImpNr | : | KSK\_IMPNR\_VL |
| Name | : | KSK\_NAME\_VL |
| Operation | : | KSK\_OPERATION\_VL |
| Sequence | : | KSK\_SEQUENCE\_VL |
| MarkerGeom | : | KSK\_MARKERGEOM\_VL |
| SegmentId | : | KSK\_KSK\_KSG\_NOR\_ID |
| Km | : | KSK\_KM\_VL |
| SectorLength | : | KSK\_SEKTORLEN\_VL |
|  | | |
|
| **MandantDetails** | **=** | **ADD\_MANDANTDET\_MSG** |
| Id | : | MAD\_ID |
| DifferenzHoehenlageSiedlungsgebiete | : | MAD\_DIFFERENZHO\_VL |
| Einwohner | : | MAD\_EINWOHNER\_VL |
| Gemeindeflaeche | : | MAD\_GEMEINDEFLA\_VL |
| MittlereHoehenlageSiedlungsgebiete | : | MAD\_MITTLEREHOE\_VL |
| Siedlungsflaeche | : | MAD\_SIEDLUNGSFL\_VL |
| Steuerertrag | : | MAD\_STEUERERTRA\_VL |
| NetzLaenge | : | MAD\_NETZLAENGE\_NR |
| IsCompleted | : | MAD\_ISCOMPLETED\_VL |
| Gemeindetyp | : | MAD\_MAD\_GEK\_NOR\_ID |
| OeffentlicheVerkehrsmittel | : | MAD\_MAD\_OVG\_NOR\_ID |
| Mandant | : | MAD\_MAD\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | MAD\_MAD\_EPR\_NOR\_ID |
|  | | |
|
| **MandantLogo** | **=** | **ADD\_MANDANTLOGO\_MSG** |
| Id | : | MAL\_ID |
| Logo | : | MAL\_LOGO\_VL |
| Height | : | MAL\_HEIGHT\_VL |
| Width | : | MAL\_WIDTH\_VL |
| Mandant | : | MAL\_MAL\_MAN\_NOR\_ID |
|  | | |
|
| **Mandant** | **=** | **ADD\_MANDANT\_MSG** |
| Id | : | MAN\_ID |
| MandantName | : | MAN\_MANDANTNAME\_VL |
| MandantBezeichnung | : | MAN\_MANDANTBEZE\_VL |
| OwnerId | : | MAN\_OWNERID\_VL |
|  | | |
|
| **MassnahmenvorschlagKatalog** | **=** | **VAT\_MASSVORSCH\_MSG** |
| Id | : | MVK\_ID |
| Typ | : | MVK\_TYP\_VL |
| DefaultKosten | : | MVK\_DEFKOSTEN\_NR |
| KatalogTyp | : | MVK\_KATALOGTYP\_VL |
| IsCustomized | : | MVK\_ISCUSTOMIZE\_VL |
| Belastungskategorie | : | MVK\_MVK\_BLK\_NOR\_ID |
| Mandant | : | MVK\_MVK\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | MVK\_MVK\_EPR\_NOR\_ID |
|  | | |
|
| **MassnahmenvorschlagTeilsystemeGIS** | **=** | **ADD\_MASSTEILGIS\_MSG** |
| Id | : | MTG\_ID |
| Projektname | : | MTG\_PROJEKTNAME\_VL |
| BezeichnungVon | : | MTG\_BEZVON\_VL |
| BezeichnungBis | : | MTG\_BEZBIS\_VL |
| Beschreibung | : | MTG\_BESCHREIBUN\_VL |
| ZustaendigeOrganisation | : | MTG\_ZUSTAENDORG\_VL |
| Teilsystem | : | MTG\_TEILSYSTEM\_VL |
| Dringlichkeit | : | MTG\_DRINGLICHKE\_VL |
| Status | : | MTG\_STATUS\_VL |
| Kosten | : | MTG\_KOSTEN\_VL |
| Laenge | : | MTG\_LAENGE\_NR |
| Shape | : | MTG\_SHAPE\_VL |
| ReferenzGruppe | : | MTG\_MTG\_RFG\_NOR\_ID |
| Mandant | : | MTG\_MTG\_MAN\_NOR\_ID |
|  | | |
|
| **NetzSummarischDetail** | **=** | **ADD\_NETZSUMDET\_MSG** |
| Id | : | NSD\_ID |
| MittlererZustand | : | NSD\_MITZST\_VL |
| Fahrbahnlaenge | : | NSD\_FBLAENGE\_NR |
| Fahrbahnflaeche | : | NSD\_FBFLAECHE\_VL |
| Belastungskategorie | : | NSD\_NSD\_BLK\_NOR\_ID |
| NetzSummarisch | : | NSD\_NSD\_NSU\_NOR\_ID |
|  | | |
|
| **NetzSummarisch** | **=** | **ADD\_NETZSUM\_MSG** |
| Id | : | NSU\_ID |
| NetzSummarischDetails | : |  |
| MittleresErhebungsJahr | : | NSU\_MITERHJAHR\_VL |
| Mandant | : | NSU\_NSU\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | NSU\_NSU\_EPR\_NOR\_ID |
|  | | |
|
| **OeffentlicheVerkehrsmittelKatalog** | **=** | **VAT\_OEFFVERKAT\_MSG** |
| Id | : | OVG\_ID |
| Typ | : | OVG\_TYP\_VL |
|  | | |
|
| **RealisierteMassnahme** | **=** | **ADD\_RELMASSTAB\_MSG** |
| Id | : | RMT\_ID |
| Projektname | : | RMT\_PROJEKTNAME\_VL |
| BezeichnungVon | : | RMT\_BEZVON\_VL |
| BezeichnungBis | : | RMT\_BEZBIS\_VL |
| Laenge | : | RMT\_LAENGE\_NR |
| BreiteFahrbahn | : | RMT\_BREITEFB\_NR |
| BreiteTrottoirLinks | : | RMT\_BREITETRL\_VL |
| BreiteTrottoirRechts | : | RMT\_BREITETRR\_VL |
| Beschreibung | : | RMT\_BESCHR\_VL |
| KostenFahrbahn | : | RMT\_FBKOSTEN\_VL |
| KostenTrottoirRechts | : | RMT\_TRRKOSTEN\_VL |
| KostenTrottoirLinks | : | RMT\_TRLKOSTEN\_VL |
| Strasseneigentuemer | : | RMT\_EIGENTUEMER\_VL |
| RealisierteMassnahmenvorschlagKatalog | : | RMT\_RMT\_RMK\_NOR\_ID |
| Belastungskategorie | : | RMT\_RMT\_BLK\_NOR\_ID |
| ErfassungsPeriod | : | RMT\_RMT\_EPR\_NOR\_ID |
| Mandant | : | RMT\_RMT\_MAN\_NOR\_ID |
|  | | |
|
| **RealisierteMassnahmeGIS** | **=** | **ADD\_REALMASSGIS\_MSG** |
| Id | : | RMG\_ID |
| BeteiligteSysteme | : |  |
| Projektname | : | RMG\_PROJEKTNAME\_VL |
| BezeichnungVon | : | RMG\_BEZVON\_VL |
| BezeichnungBis | : | RMG\_BEZBIS\_VL |
| Laenge | : | RMG\_LAENGE\_NR |
| BreiteFahrbahn | : | RMG\_BREITEFB\_NR |
| BreiteTrottoirLinks | : | RMG\_BREITETRL\_VL |
| BreiteTrottoirRechts | : | RMG\_BREITETRR\_VL |
| KostenGesamtprojekt | : | RMG\_KOSTGESAMT\_VL |
| KostenFahrbahn | : | RMG\_KOSTENFB\_VL |
| KostenTrottoirLinks | : | RMG\_KOSTENTRL\_VL |
| KostenTrottoirRechts | : | RMG\_KOSTENTRR\_VL |
| Strasseneigentuemer | : | RMG\_EIGENTUEMER\_VL |
| Beschreibung | : | RMG\_BESCHREIBUN\_VL |
| LeitendeOrganisation | : | RMG\_LEITENDEORG\_VL |
| Shape | : | RMG\_SHAPE\_VL |
| MassnahmenbeschreibungFahrbahn | : | RMG\_RMG\_RMK\_NOR\_ID |
| Belastungskategorie | : | RMG\_RMG\_BLK\_NOR\_ID |
| ReferenzGruppe | : | RMG\_RMG\_RFG\_NOR\_ID |
| Mandant | : | RMG\_RMG\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | RMG\_RMG\_EPR\_NOR\_ID |
|  | | |
|
| **RealisierteMassnahmenvorschlagKatalog** | **=** | **VAT\_RMASSVOR\_MSG** |
| Id | : | RMK\_ID |
| Typ | : | RMK\_TYP\_VL |
| IsCustomized | : | RMK\_ISCUSTOMIZE\_VL |
| Mandant | : | RMK\_RMK\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | RMK\_RMK\_EPR\_NOR\_ID |
|  | | |
|
| **RealisierteMassnahmeSummarsich** | **=** | **ADD\_RELMASSSUM\_MSG** |
| Id | : | RMS\_ID |
| Beschreibung | : | RMS\_BESCHR\_VL |
| KostenFahrbahn | : | RMS\_FBKOSTEN\_VL |
| Projektname | : | RMS\_PROJEKTNAME\_VL |
| Fahrbahnflaeche | : | RMS\_FBFLAECHE\_VL |
| Strasseneigentuemer | : | RMS\_EIGENTUEMER\_VL |
| ErfassungsPeriod | : | RMS\_RMS\_EPR\_NOR\_ID |
| Mandant | : | RMS\_RMS\_MAN\_NOR\_ID |
| Belastungskategorie | : | RMS\_RMS\_BLK\_NOR\_ID |
|  | | |
|
| **ReferenzGruppe** | **=** | **ADR\_REFGRUPPE\_MSG** |
| Id | : | RFG\_ID |
| AchsenReferenzen | : |  |
| StrassenabschnittGISList | : |  |
| ZustandsabschnittGISList | : |  |
| KoordinierteMassnahmeGISList | : |  |
| MassnahmenvorschlagTeilsystemeGISList | : |  |
| RealisierteMassnahmeGISList | : |  |
| CopiedFrom | : | RFG\_RFG\_RFG\_NOR\_ID |
|  | | |
|
| **Schadendetail** | **=** | **ADD\_SCHADDET\_MSG** |
| Id | : | SCD\_ID |
| SchadendetailTyp | : | SCD\_DETAILTYP\_VL |
| SchadenschwereTyp | : | SCD\_SCHWERETYP\_VL |
| SchadenausmassTyp | : | SCD\_AUSMASSTYP\_VL |
|  | | |
|
| **Schadengruppe** | **=** | **ADD\_SCHADGRUPPE\_MSG** |
| Id | : | SCG\_ID |
| SchadengruppeTyp | : | SCG\_GRUPPETYP\_VL |
| SchadenschwereTyp | : | SCG\_SCHWERETYP\_VL |
| SchadenausmassTyp | : | SCG\_AUSMASSTYP\_VL |
|  | | |
|
| **Sektor** | **=** | **SRD\_SEKTOR\_MSG** |
| Id | : | SEK\_ID |
| BsId | : | SEK\_BSID\_VL |
| Km | : | SEK\_KM\_VL |
| SectorLength | : | SEK\_SEKTORLEN\_VL |
| Name | : | SEK\_NAME\_VL |
| Sequence | : | SEK\_SEQUENCE\_VL |
| MarkerGeom | : | SEK\_MARKERGEOM\_VL |
| Operation | : | SEK\_OPERATION\_VL |
| ImpNr | : | SEK\_IMPNR\_VL |
| AchsenSegment | : | SEK\_SEK\_ACS\_NOR\_ID |
| CopiedFrom | : | SEK\_SEK\_SEK\_NOR\_ID |
|  | | |
|
| **StrassenabschnittGIS** | **=** | **ADD\_STRAGIS\_MSG** |
| Id | : | STG\_ID |
| Zustandsabschnitten | : |  |
| InspektionsRtStrAbschnitte | : |  |
| Strassenname | : | STG\_STRASSENNAM\_VL |
| BezeichnungVon | : | STG\_BEZVON\_VL |
| BezeichnungBis | : | STG\_BEZBIS\_VL |
| Belag | : | STG\_BELAG\_VL |
| Laenge | : | STG\_LAENGE\_NR |
| LaengenkorrekturAnfang | : | STG\_LAKORRANFANG\_NR |
| LaengenkorrekturEnde | : | STG\_LAKORRENDE\_NR |
| BreiteFahrbahn | : | STG\_BREITEFB\_NR |
| Trottoir | : | STG\_TROTTOIR\_VL |
| BreiteTrottoirLinks | : | STG\_BREITETRL\_VL |
| BreiteTrottoirRechts | : | STG\_BREITETRR\_VL |
| Strasseneigentuemer | : | STG\_EIGENTUEMER\_VL |
| Ortsbezeichnung | : | STG\_ORTSBEZ\_VL |
| Shape | : | STG\_SHAPE\_VL |
| IsLocked | : | STG\_ISLOCKED\_VL |
| Belastungskategorie | : | STG\_STG\_BLK\_NOR\_ID |
| Mandant | : | STG\_STG\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | STG\_STG\_EPR\_NOR\_ID |
| ReferenzGruppe | : | STG\_STG\_RFG\_NOR\_ID |
| CopiedFrom | : | STG\_STG\_STG\_NOR\_ID |
|  | | |
|
| **Strassenabschnitt** | **=** | **ADD\_STRATAB\_MSG** |
| Id | : | STT\_ID |
| Zustandsabschnitten | : |  |
| Strassenname | : | STT\_STRASSENNAM\_VL |
| BezeichnungVon | : | STT\_BEZVON\_VL |
| BezeichnungBis | : | STT\_BEZBIS\_VL |
| Belag | : | STT\_BELAG\_VL |
| Laenge | : | STT\_LAENGE\_NR |
| BreiteFahrbahn | : | STT\_BREITEFB\_NR |
| Trottoir | : | STT\_TROTTOIR\_VL |
| BreiteTrottoirLinks | : | STT\_BREITETRL\_VL |
| BreiteTrottoirRechts | : | STT\_BREITETRR\_VL |
| Strasseneigentuemer | : | STT\_EIGENTUEMER\_VL |
| Ortsbezeichnung | : | STT\_ORTSBEZ\_VL |
| Abschnittsnummer | : | STT\_ABSCHNITTSNR\_VL |
| Belastungskategorie | : | STT\_STT\_BLK\_NOR\_ID |
| Mandant | : | STT\_STT\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | STT\_STT\_EPR\_NOR\_ID |
|  | | |
|
| **TestUserInfo** | **=** | **ADP\_TESTUSER\_MSG** |
| Id | : | TUI\_ID |
| UserName | : | TUI\_USERNAME\_VL |
| Mandator | : | TUI\_MANDATOR\_VL |
| Rolle | : | TUI\_ROLLE\_VL |
|  | | |
|
| **WiederbeschaffungswertKatalog** | **=** | **VAT\_WBBKAT\_MSG** |
| Id | : | WBK\_ID |
| GesamtflaecheFahrbahn | : | WBK\_FLAEGESFB\_NR |
| FlaecheFahrbahn | : | WBK\_FLAECHEFB\_NR |
| FlaecheTrottoir | : | WBK\_FLAECHETR\_NR |
| Gesamtlaenge | : | WBK\_GESAMTLAENG\_NR |
| GesamtflaecheFahrbahnTrottoir | : | WBK\_FALEGESFBTR\_NR |
| AlterungsbeiwertI | : | WBK\_ALTERUNGI\_NR |
| AlterungsbeiwertII | : | WBK\_ALTERUNGII\_NR |
| IsCustomized | : | WBK\_ISCUSTOMIZE\_VL |
| Belastungskategorie | : | WBK\_WBK\_BLK\_NOR\_ID |
| Mandant | : | WBK\_WBK\_MAN\_NOR\_ID |
| ErfassungsPeriod | : | WBK\_WBK\_EPR\_NOR\_ID |
|  | | |
|
| **ZustandsabschnittGIS** | **=** | **ADD\_ZSTGIS\_MSG** |
| Id | : | ZSG\_ID |
| Schadengruppen | : |  |
| Schadendetails | : |  |
| Zustandsindex | : | ZSG\_ZUSTANDSIND\_NR |
| BezeichnungVon | : | ZSG\_BEZVON\_VL |
| BezeichnungBis | : | ZSG\_BEZBIS\_VL |
| Erfassungsmodus | : | ZSG\_ERFASSUNGSM\_VL |
| Laenge | : | ZSG\_LAENGE\_NR |
| Aufnahmedatum | : | ZSG\_AUFNAHMEDAT\_DT |
| Aufnahmeteam | : | ZSG\_AUFNAHMETEA\_VL |
| Wetter | : | ZSG\_WETTER\_VL |
| Bemerkung | : | ZSG\_BEMERKUNG\_VL |
| ZustandsindexTrottoirLinks | : | ZSG\_ZSTINDTRL\_VL |
| ZustandsindexTrottoirRechts | : | ZSG\_ZSTINDTRR\_VL |
| KostenMassnahmenvorschlagFahrbahn | : | ZSG\_FBKOSTEN\_VL |
| DringlichkeitFahrbahn | : | ZSG\_FBDRINGLICH\_VL |
| KostenMassnahmenvorschlagTrottoirRechts | : | ZSG\_TRRKOSTEN\_VL |
| DringlichkeitTrottoirRechts | : | ZSG\_TRRDRINGLIC\_VL |
| KostenMassnahmenvorschlagTrottoirLinks | : | ZSG\_TRLKOSTEN\_VL |
| DringlichkeitTrottoirLinks | : | ZSG\_TRLDRINGLIC\_VL |
| Shape | : | ZSG\_SHAPE\_VL |
| MassnahmenvorschlagFahrbahn | : | ZSG\_ZSG\_MVK\_FB\_ID |
| MassnahmenvorschlagTrottoirRechts | : | ZSG\_ZSG\_MVK\_TRR\_ID |
| MassnahmenvorschlagTrottoirLinks | : | ZSG\_ZSG\_MVK\_TRL\_ID |
| StrassenabschnittGIS | : | ZSG\_ZSG\_STG\_NOR\_ID |
| ReferenzGruppe | : | ZSG\_ZSG\_RFG\_NOR\_ID |
| CopiedFrom | : | ZSG\_ZSG\_ZSG\_NOR\_ID |
|  | | |
|
| **Zustandsabschnitt** | **=** | **ADD\_ZSTTAB\_MSG** |
| Id | : | ZST\_ID |
| Schadengruppen | : |  |
| Schadendetails | : |  |
| Zustandsindex | : | ZST\_ZSTIND\_NR |
| Erfassungsmodus | : | ZST\_ERFASSUNGSM\_VL |
| BezeichnungVon | : | ZST\_BEZVON\_VL |
| BezeichnungBis | : | ZST\_BEZBIS\_VL |
| Laenge | : | ZST\_LAENGE\_NR |
| Aufnahmedatum | : | ZST\_AUFNDAT\_DT |
| Aufnahmeteam | : | ZST\_AUFNTEAM\_VL |
| Wetter | : | ZST\_WETTER\_VL |
| Bemerkung | : | ZST\_BEMERKUNG\_VL |
| ZustandsindexTrottoirLinks | : | ZST\_ZSTINDTRL\_VL |
| ZustandsindexTrottoirRechts | : | ZST\_ZSTINDTRR\_VL |
| KostenMassnahmenvorschlagFahrbahn | : | ZST\_FBKOSTEN\_VL |
| DringlichkeitFahrbahn | : | ZST\_FBDRINGLICH\_VL |
| KostenMassnahmenvorschlagTrottoirRechts | : | ZST\_TRRKOSTEN\_VL |
| DringlichkeitTrottoirRechts | : | ZST\_TRRDRINGLIC\_VL |
| KostenMassnahmenvorschlagTrottoirLinks | : | ZST\_TRLKOSTEN\_VL |
| DringlichkeitTrottoirLinks | : | ZST\_TRLDRINGLIC\_VL |
| MassnahmenvorschlagFahrbahn | : | ZST\_ZST\_MVK\_FB\_ID |
| MassnahmenvorschlagTrottoirRechts | : | ZST\_ZST\_MVK\_TRR\_ID |
| MassnahmenvorschlagTrottoirLinks | : | ZST\_ZST\_MVK\_TRL\_ID |
| Strassenabschnitt | : | ZST\_ZST\_STT\_NOR\_ID |
|  | | |
|

### Physisches Datenmodell



Anhang: Physisches Datenbankmodell

# Anhang

## Datenmodell EMSG-Master



Anhang: Datenmodell EMSG-Master

1. Fachdaten wird als Synonym zu Spezialistendaten aus dem Pflichtenheft verwendet. [↑](#footnote-ref-2)
2. Mit eine Duplikation der Logik ist vor allem bei der clientseitigen Validierung zu rechnen [↑](#footnote-ref-3)
3. Unter dem Begriff „Achsen“ und „Achsdaten“ werden alle Entitäten gemeint, die eine Achse bilden. Die wichtigsten sind „Achse“, „Achssegment“ und „Sektor“. [↑](#footnote-ref-4)
4. siehe auch 4.4.8.1 für eine Erklärung des Zusammenhangs zwischen Achsen und fachspezifischen EMGS GIS-Daten [↑](#footnote-ref-5)
5. Diese Intervalle werden vom Basissystem vorgegeben, der Achsen Import Service prüft jede Stunde zu Minute 0 und 30 auf neu hinzugefügte Dateien. [↑](#footnote-ref-6)
6. In Folge wird für diese Tabellen der Begriff „Kopie Achsdaten“ verwendet. [↑](#footnote-ref-7)
7. Der Prozess stellt damit einen Job, der durch einen Scheduler regelmässig gestartet wird, dar. Das Intervall ist konfigurierbar. [↑](#footnote-ref-8)
8. Die Zustandsabschnitte werden bei der Achsenaktualisierung vom System gelöscht, falls sich der dazugehörige Strassenabschnitt nicht automatisch nachführen lässt und folglich eine Benutzerinteraktion zur Neudefinition des Strassenabschnitts nötig ist. Das Löschen der Zustandsabschnitte ist nötig, um die geometrische Konsistenz zwischen Zustandsabschnitten und den geänderten Strassenabschnitten zu gewährleisten. [↑](#footnote-ref-9)
9. Der Checkin/Checkout-Mechanismus wird beim Export der Daten auf EMSG-Mobile verwendet (vgl. Kap. 5.1.2). Diese Voraussetzung bedingt, dass es zu keinen Inkonsistenzen der Achsdaten zwischen EMSG-Mobile und EMSG-Master kommen kann. Das System prüft vor dem Start des Prozesses, ob Check-Outs für den entsprechenden Mandanten vorhanden sind und gibt in diesem Falle eine Fehlermeldung aus. [↑](#footnote-ref-10)
10. Dies gilt nicht für den initialen Achsenimport [↑](#footnote-ref-11)
11. Damit werden Fälle bezeichnet, welche die Datenkonsistenz potentiell gefährden und damit zu Inkonsistenzen führen können. [↑](#footnote-ref-12)
12. Eine technische Detaillierung und prototypische Umsetzung, bzw. erste Tests sind für die erste Iteration der FA EMSG vorgesehen. [↑](#footnote-ref-13)
13. Quelle: [6] (Kapitel 5) [↑](#footnote-ref-14)
14. Die Transferdefinition beinhaltet Optionen, wie, wann und wohin die Daten aus dem Basissystem exportiert werden – Details zu Transferdefinition findet man unter [6] (Kapitel 5). [↑](#footnote-ref-15)
15. Streng genommen handelt es sich um eine Version, da diese „Kopie“ von jener des Basissystems divergieren kann. Der Verständlichkeit halber wird dennoch der Begriff Kopie verwendet. [↑](#footnote-ref-16)
16. Die Implementierung dieses Prozesses wird in Form eines Windows Services durchgeführt. [↑](#footnote-ref-17)
17. Dieses signalisiert, ob ein bestimmter Datensatz neu angelegt, aktualisiert oder gelöscht wurde (INSERT, UPDATE, DELETE). [↑](#footnote-ref-18)
18. Quelle: [6] (Kapitel 5) [↑](#footnote-ref-19)
19. Ein Eintrag in die Tabelle ACHSUPDATECONFLICT signalisiert, dass der Benutzer auf einem neuen Achsensegment noch Strassenabschnitte und Zustände definieren muss. Dies jedoch nur, wenn das Achssegment über einen inkrementellen Achsen-Import neu hinzugekommen ist. [↑](#footnote-ref-20)
20. Da ein Update in diesem Fall eine komplette, jedoch neue Geometrie beinhaltet wird in diesem Fall keine Warnung erzeugt; Für den Benutzer ergeben sich hiermit keine weiteren Schritte. Die Achse wird als „Importiert“ markiert. [↑](#footnote-ref-21)
21. Die Geometrie der Achse wird als „Gelöscht“ in den Achsaktualisierungen dargestellt. [↑](#footnote-ref-22)
22. http://de.wikipedia.org/wiki/Martin-Notation [↑](#footnote-ref-23)
23. Redundante direkte Verbindungen wurden aus Performance-Gründen hinzugefügt. [↑](#footnote-ref-24)
24. Zur einfacheren und besseren Benennung der Tabellen tragen die Tabellennamen das Suffix **Tabellarisch**. [↑](#footnote-ref-25)
25. Achsdaten werden über die INTERLIS2 Schnittstelle in die Fachapplikation EMSG importiert, bzw. aktualisiert. Diese Daten werden nicht durch Benutzerinteraktion im der EMSG-Fachapplikation manipuliert. (Der Ausnahmefall ist der Anwendungsfall für die Aktualisierung der Achsen). [↑](#footnote-ref-26)
26. Der Query Layer ist seit GeoServer verfügbar und erlaubt den Zugriff auf Geodaten ohne die Notwendigkeit einer Layer Definition, welche für den Zugriff via Direct Connect weiterhin notwendig ist. [↑](#footnote-ref-27)
27. Die n:m Assoziation zwischen Strassenabschnitten und Inspektionsrouten, welche derzeit noch nicht notwendig wäre, wird aus Gründen der Flexibilität gewählt. [↑](#footnote-ref-28)
28. Mit SpecFlow werden Anforderungen in einer für den Analysten lesbaren Form (Gherkin) beschrieben, welche die Geschäftslogik einbinden und verwenden und in Folge Akzeptanzkriterien verifizieren können. [↑](#footnote-ref-29)
29. Entwicklerlizenz kostenpflichtig, Laufzeitumgebung kostenfrei [↑](#footnote-ref-30)
30. Entwicklerlizenz kostenpflichtig, Laufzeitumgebung kostenfrei [↑](#footnote-ref-31)
31. http://resx-localization-studio.net/ [↑](#footnote-ref-32)
32. http://notepad-plus-plus.org/ [↑](#footnote-ref-33)
33. Prinzipiell sind die Geometrien der EMSG Fachdaten abgeleitet aus den EMSG Achsen und damit redudant. Dies ist aber aus Performancegründen notwendig. [↑](#footnote-ref-34)
34. Jedes Datenpaket beinhaltet eine Liste mit den EMSG-Mobile Softwareversionen mit denen es Kompatible ist. Ist die aktuelle Softwareversion nicht der Liste enthalten, so kann das Datenpaket nicht importiert werden. [↑](#footnote-ref-35)
35. Die Entscheidung, wann der Bearbeitungsvorgang abgeschlossen ist obliegt dem EMS Mobile Benutzer. [↑](#footnote-ref-36)
36. Die anzuzeigenden WMS Layer wurden von der LR vorgegeben, können aber noch ohne Zusatzaufwand vor der Implementierung adaptiert werden. [↑](#footnote-ref-37)
37. Als funktionaler Anwendungsfall wurde das Bearbeiten einen Linienelementes auf der Karte, sowie dessen Erzeugung (Digitalisierung auf der Karte) inkl. Eingabe und Manipulation der zugehörigen Attribute definiert und umgesetzt. [↑](#footnote-ref-38)
38. http://nuget.org/ [↑](#footnote-ref-39)
39. <http://visualstudiogallery.msdn.microsoft.com/c255a1e4-04ba-4f68-8f4e-cd473d6b971f> [↑](#footnote-ref-40)
40. <http://www.asp.net/mvc/mvc3> [↑](#footnote-ref-41)
41. <http://learn.iis.net/page.aspx/868/iis-express-overview/> [↑](#footnote-ref-42)
42. https://addons.mozilla.org/de/firefox/addon/firebug [↑](#footnote-ref-43)
43. http://www.jetbrains.com/resharper/ [↑](#footnote-ref-44)
44. http://www.testdriven.net [↑](#footnote-ref-45)
45. http://www.nunit.org/ [↑](#footnote-ref-46)