

WEGA-GDM-Server-Informationen

Copyright© 1997-2002

M.O.S.S. Computer Grafik Systeme GmbH

Postanschrift: Hohenbrunner Weg 13, 82024 Taufkirchen

Telefon: (089) 666 75 100 Fax: 666 75 180

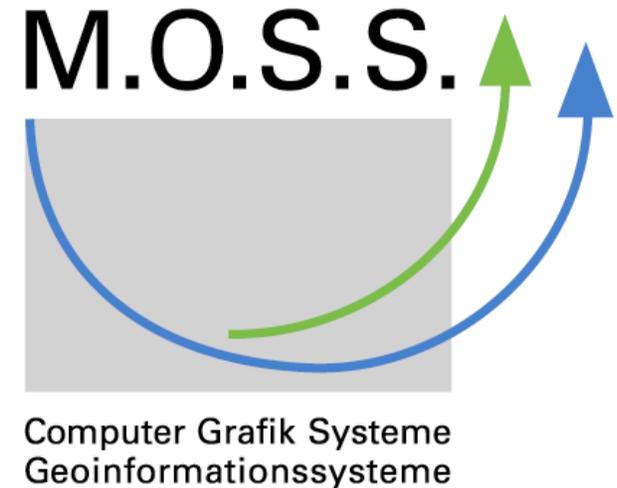
Geschäftsführer: Dipl.-Math. Hans Braun

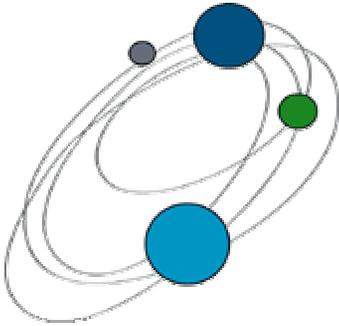
HR B: 82009 Amtsgericht-Registergericht-München

Alle Rechte vorbehalten

M.O.S.S. behält sich das Recht vor, Änderungen an den Spezifikationen und anderen Inhalten der Publikation ohne vorherige Ankündigung vorzunehmen.

Diese Publikation darf ohne Zustimmung von M.O.S.S. nicht kopiert werden und ist für die alleinige Benutzung durch M.O.S.S.-Kunden bestimmt.





WEGA-GDM steht für Web Enabled Graphics Architecture Geo Dokumenten Management System.

Das Dokumentenverwaltungssystem archiviert applikationsübergreifend *Dokumente mit Ortsbezug*. Für jeden Dokumententyp kann über WEGA-GDM das gewünschte Anzeige- bzw. Bearbeitungsprogramm festgelegt und dann automatisch gestartet werden.

WEGA-GDM basiert auf einer *Client-Server-Architektur* und erlaubt die Verwaltung von beliebigen Dokumenttypen. Der Zugriff auf die Dokumente erfolgt über Sachdaten oder über die Grafik. Es können Dokumente archiviert, ausgewählt, dargestellt und bearbeitet werden, z.B. Grafikdokumente (Raster- und Vektorgrafik), Textdokumente (z.B. MS-Office Dokumente) oder auch Web-Seiten.

Gemeinsames Ordnungskriterium aller Dokumente ist der Ortsbezug, d.h. alle Dokumente sind über Koordinaten verortet und so selektierbar.

Darüber hinaus sind die Dokumente mit applikationsabhängigen Zusatzinformationen (Sachdaten) hinterlegt.

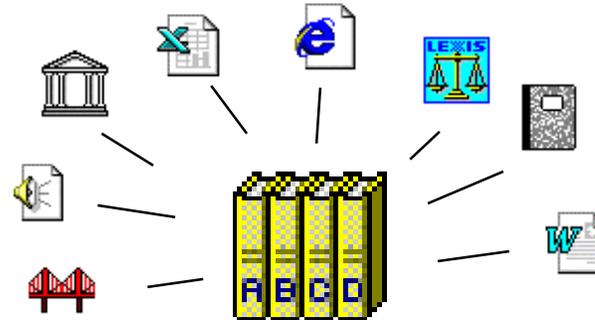
Die Dokumente können so gezielt auf zwei Arten ausgewählt werden: Grafisch-räumlich über eine Karte oder inhaltsorientiert über eine Sachdatenabfrage.

WEGA-GDM ist somit eine Dokumentenmanagementlösung, die nicht nur in Bereichen des Vermessungswesens sondern auch in Abteilungen der technischen Dokumentation von Energieversorgern und Kommunen genutzt wird.

GDM – Integration von Karten und Rissen

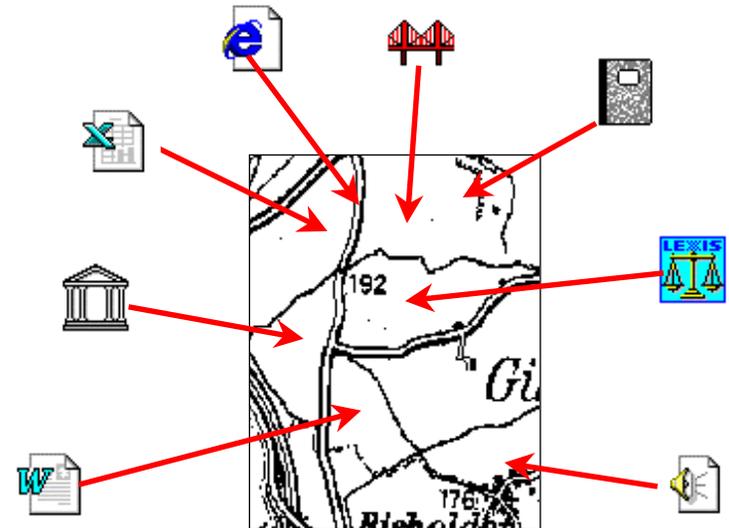
Ausgangspunkt: Große Anzahl unterschiedlicher Dokumente

- Bauanträge
- Genehmigungsunterlagen
- Vermessungsunterlagen
- Schadensberichte
- Bilder
- Multimedia
- Filme, Töne
- HTML-Dokumente
- ...



Gemeinsamkeit: Ortsbezug

Ziel: Dokumente nach Ortsbezug suchen



GDM – Dokumenttyp: NivP-Beschreibung

Landesvermessungsamt
 Rheinland-Pfalz

NivP-Beschreibung

27 1 5677
 Kartellnummer TK 25 Nr.

Bez. i. i. D.: 20 1 70

NivP (1) Name: MB Koblenz,
Oberpostdirection

Art der Festlegung: Tonnenförmiger
Eisenbolzen, Inschrift HP

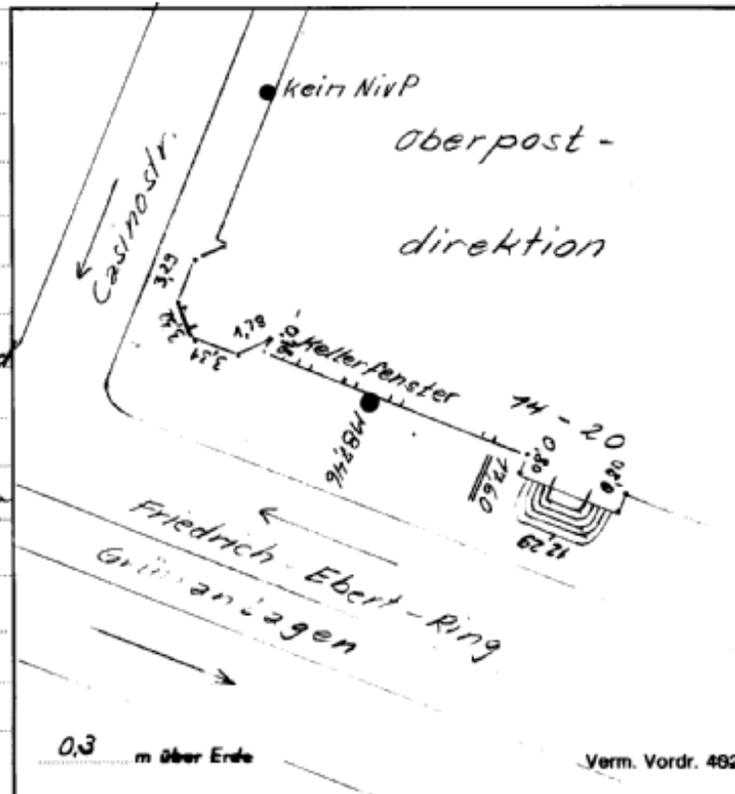
Standfestigkeit (Geologie): 21313

Eigentümer: Bundesrepublik Deutschland
Deutsche Bundespost

Festgelegt durch: RFL Jahr: 1938

Eingemessen von: Görgen VA am: 30.3.92

Bemerkungen: Rosengitter vor Bolzen.
Gitter kann abgeschraubt werden.
Bauer 24.11.86



GDM – Dokumenttyp: TP-Beschreibung

Landesvermessungsamt
Rheinland-Pfalz

TP-Beschreibung

13 / **5611**
Karteinummer Top Karte 1:25.000

TP (3) Punktname: Oberlahnstein, Burg Lahnneck

Bezeichnung im Liegenschaftskataster

Lageskizze (nach Norden orientiert)

Bezeichnung im Liegenschaftskataster

Katasteramt: St. Goarshausen

Anschlussrichtung: TP ()

Kreis: Rhein-Lahn

Gemeinde: Lahnstein



Gemarkung: Oberlahnstein

Eigenümer der Marksteinflächen (Flur:)

Flurstück:)

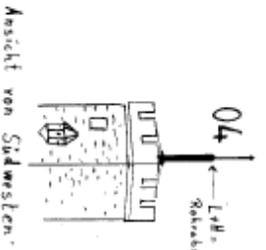
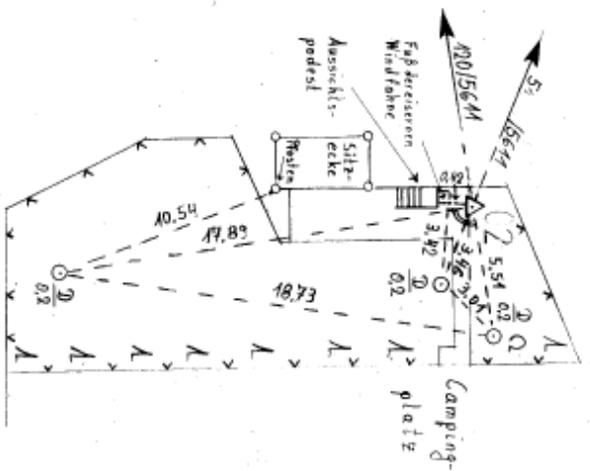
Eigentümer des umgebenden Grundstücks:

(Flur: 46 Flurstück: 34/13 Nutzungstyp:)

Erkennungsschicht: Grenzflüsse, Röhrl, Mischflüsse, Kernflüsse, Mischflüsse

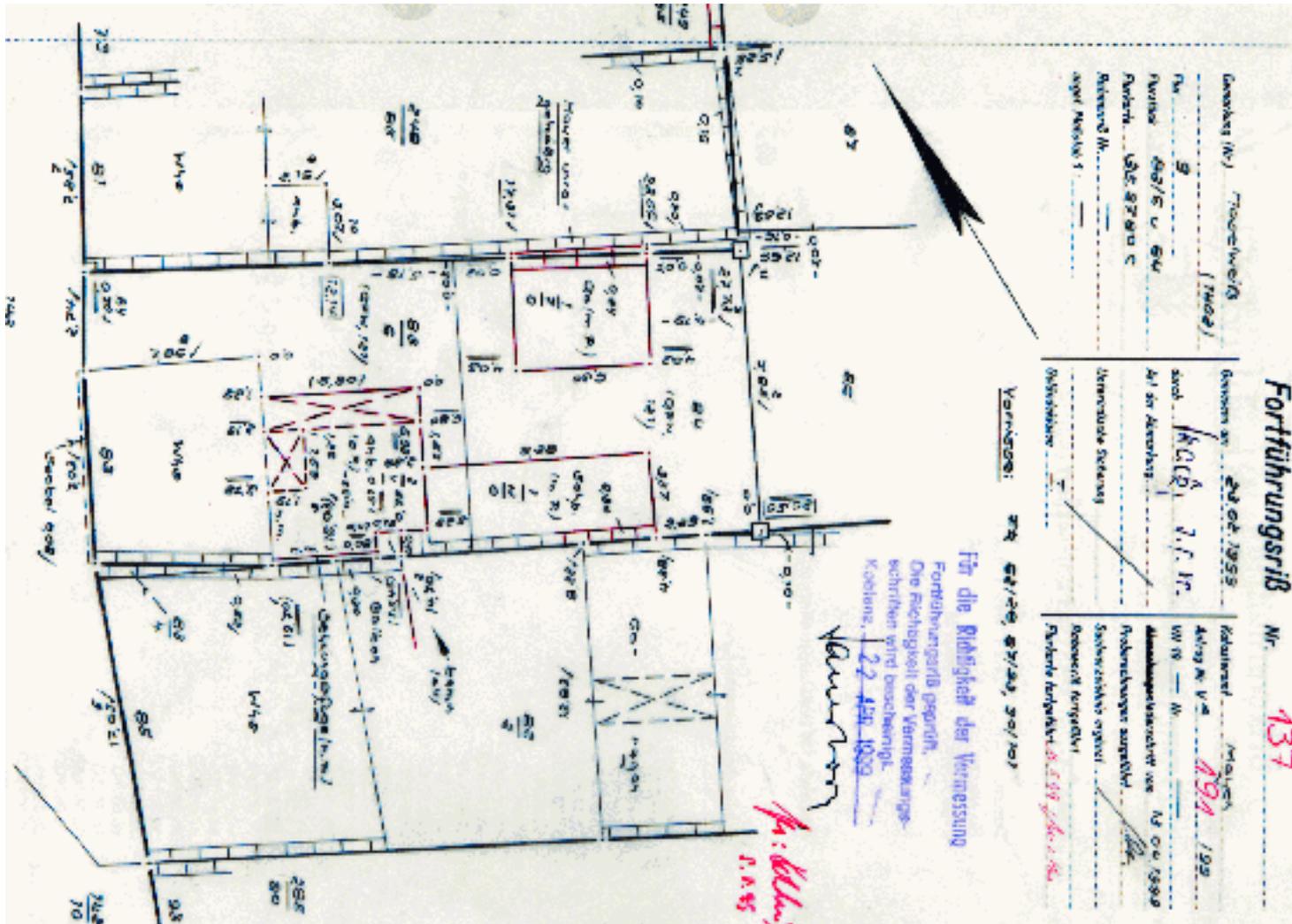
Wahlort: Prassels, Rädger, Prassen, Syron, Prassen, Jüngel, Prassen

Einmessungsskizze (Überzeichnung ist durch Pfeil angedeutet)



Sicherungsmasse beziehen sich auf das Plattenkreuz; gemessen von: Gerz VO1 am: 10.03.1989

GDM – Dokumenttyp: Fortführungsriß



GDM – Dokumenttyp: SFP-Beschreibung

SFP-Beschreibung

Landesvermessungsamt
Rheinland-Pfalz

SFP (3) Name: B. Stolzenfels
Rheinher Straße 1

Art der Festlegung: Zylinderförmiger
Gußblocken, Inschrift Ht ϕ 60 mm

Standortskizze (Geolog): 313/2

Eigenname: Taser Heinrich u. Sibylla

Festgelegt durch: DB Jahr: —

Eingemessen von: Migretz VHS am: 14.1996

Bemerkungen: 50cm über B ein Briefkasten

Höheabweichung (Festlegung-Aufnahmehöhe):
 Δh (B über Boden) = -0,92 m

Qualität als SFP: gut

52 5611

Bezeichnung im inneren Dienst: R85/30T
Kartellnummer Top Karte 1: 25000

Ident TP — Bez L L D 9152
MWP 234

Koblenz
Rheinher Straße n. Hain
17,18
901,8
55
351

0,92 m über Erde

Dauer Besondereinstellung

22.10.1985

↖ Richtung der
Rohwinkelmessung
in der Skizze

19/85

GDM – Dokumenttyp: Foto

Foto zur Dokumentation der Lage eines Schwerefestpunkts (SFP)



GDM – Anforderungen

- Große Anzahl Dokumente (in der Praxis > 100.000 Dokumente)
- Zentraler Zugriff auf die Dokumente
- Selektiver *Zugriff* anhand von
 - *geometrischer* Information (Punkt, Rechteck, Polygon, ...)
 - *Sachdaten* (Zeitpunkt der Erstellung, Zuständigkeit, Katasteramt, ...)
- Anbindung von (vorhandenen) Windows-PC's
- Nutzung existierender Software zur Visualisierung und Bearbeitung:
 - Office-Produkte (z.B. WinWord),
 - Grafik-Applikationen (z.B. RoSy, ImageViewer, AutoCAD)
- Zugriffskontrolle:
 - Zugangsbeschränkungen, Protokollierung, Abrechnung, ...

WEGA-GDM

bestehend aus

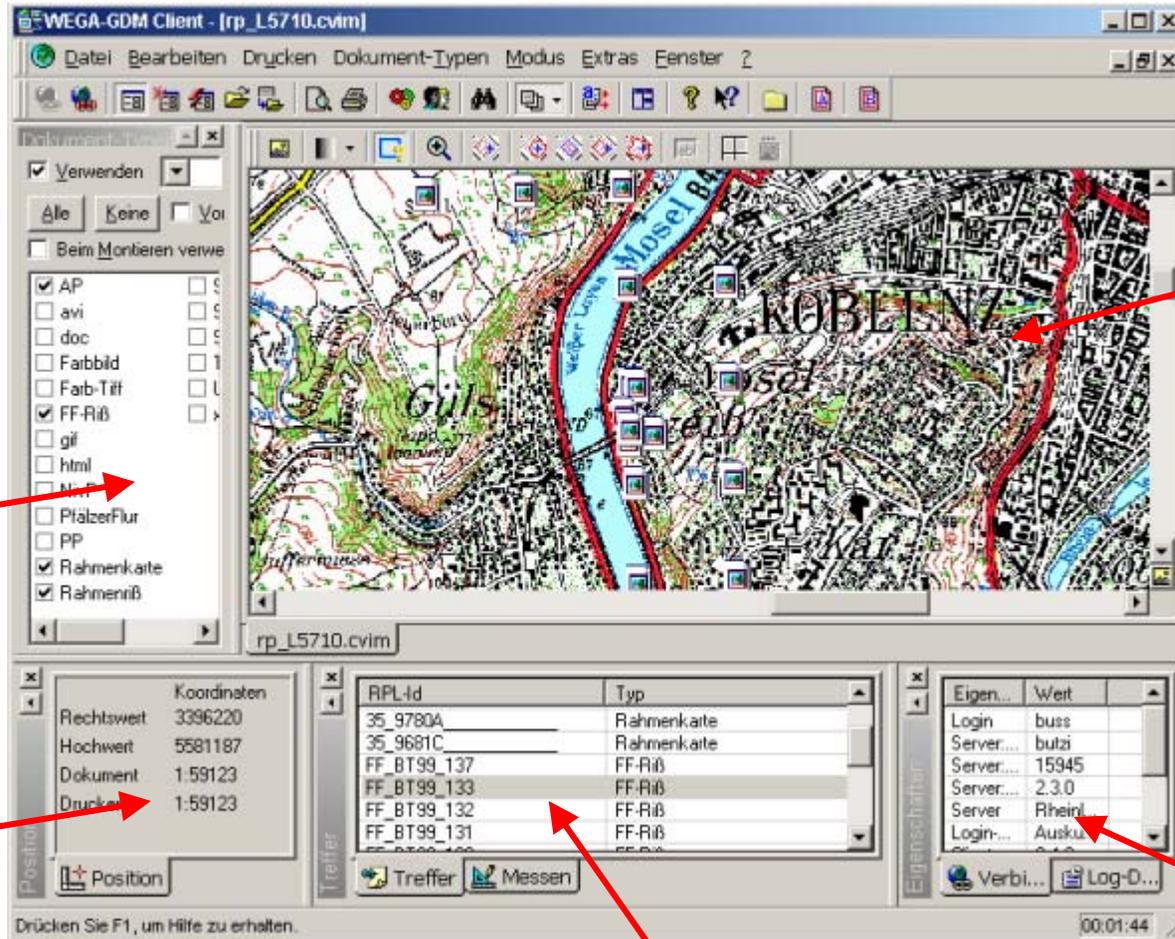
GDM-Server

- bedient alle Anfragen,
- verwaltet alle Dokumente,
- überwacht die Zugriffe,
- kontrolliert den Datenbestand (Datenbank, Dateisystem, ...)
- definiert ein eigenes Protokoll für die Clientkommunikation (kein HTTP).

GDM-Client

- regelt die Server-Kommunikation,
- steuert Zugriff auf Dokumente (Cache, Datenaustausch mit dem Server),
- aktiviert die zugeordneten Visualisierungs- bzw. Bearbeitungsprogramme,
- erlaubt die Einbringung neuer Dokumente in den zentralen Datenbestand,
- regelt die Kommunikation mit VB-Applikationsmodulen (Benutzerverwaltung, kundenspezifische Formulare, ...).

GDM – GDM-Client-GUI



Fenster zum Auswählen der Dokumenttypen

Dokumentviewer

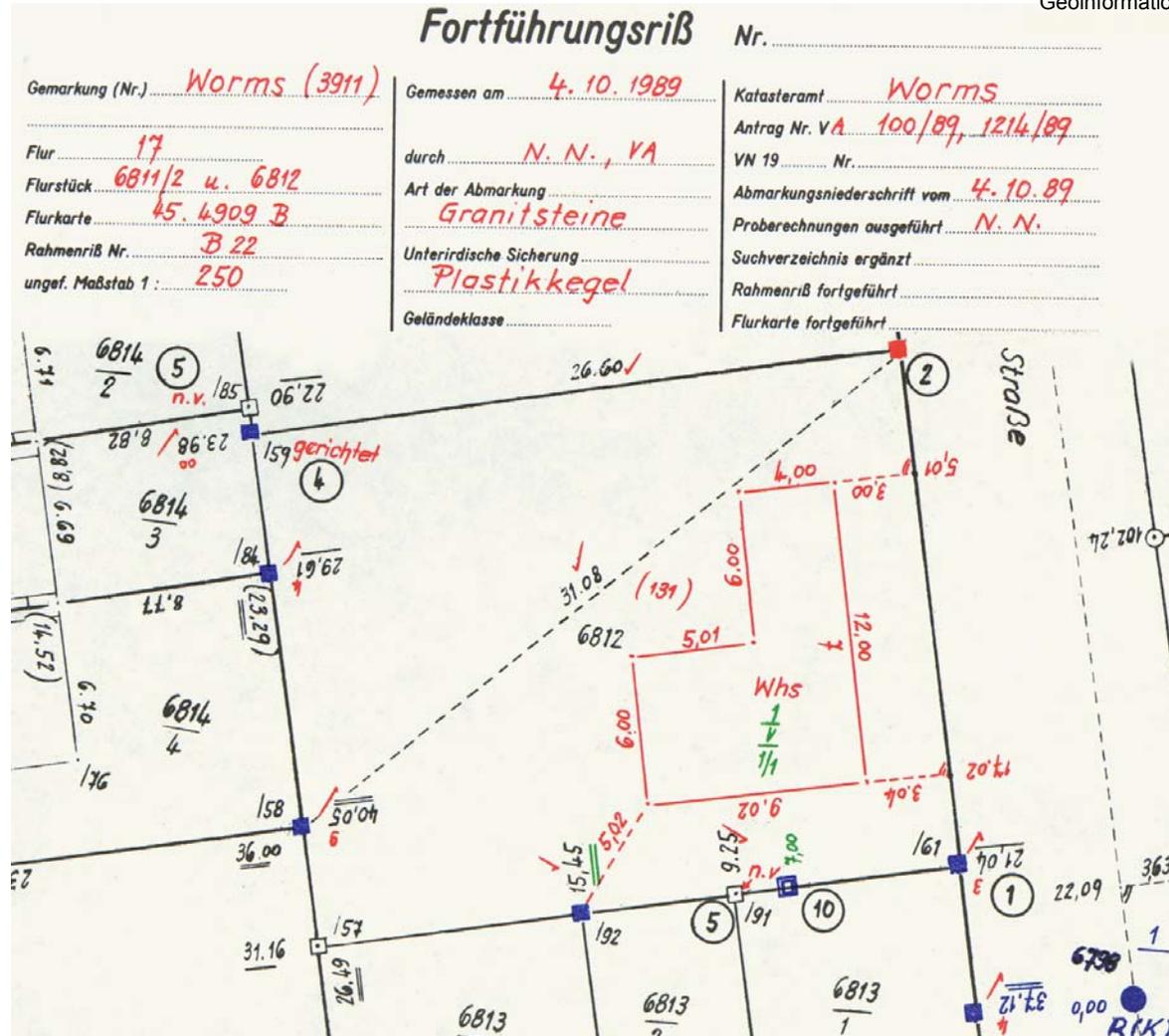
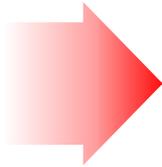
Koordinateninformation

Liste aller gefundenen Dokumente = Trefferliste

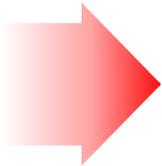
Protokollierung und Verbindungsinformationen

GDM – Trennung von Sachdaten und Grafik

Sach-
daten



Grafik



GDM – Eingabe der Sachdaten per Formular

Art des Risses:
 Gemarkungsschlüssel:
 Flur: Seitennummer: ()
 Blatt: von bis: Scan Datum:
 Jahrgang der Übernahme: Vorläufiger Nachweis:

Veränderte Flurstücke:

Gemarkung	Flur	Blatt von		bis			
		Zähler	/	Nenner	Zähler	/	Nenner
1							

Entstandene Flurstücke:

Gemarkung	Flur	Blatt von		bis			
		Zähler	/	Nenner	Zähler	/	Nenner
1							

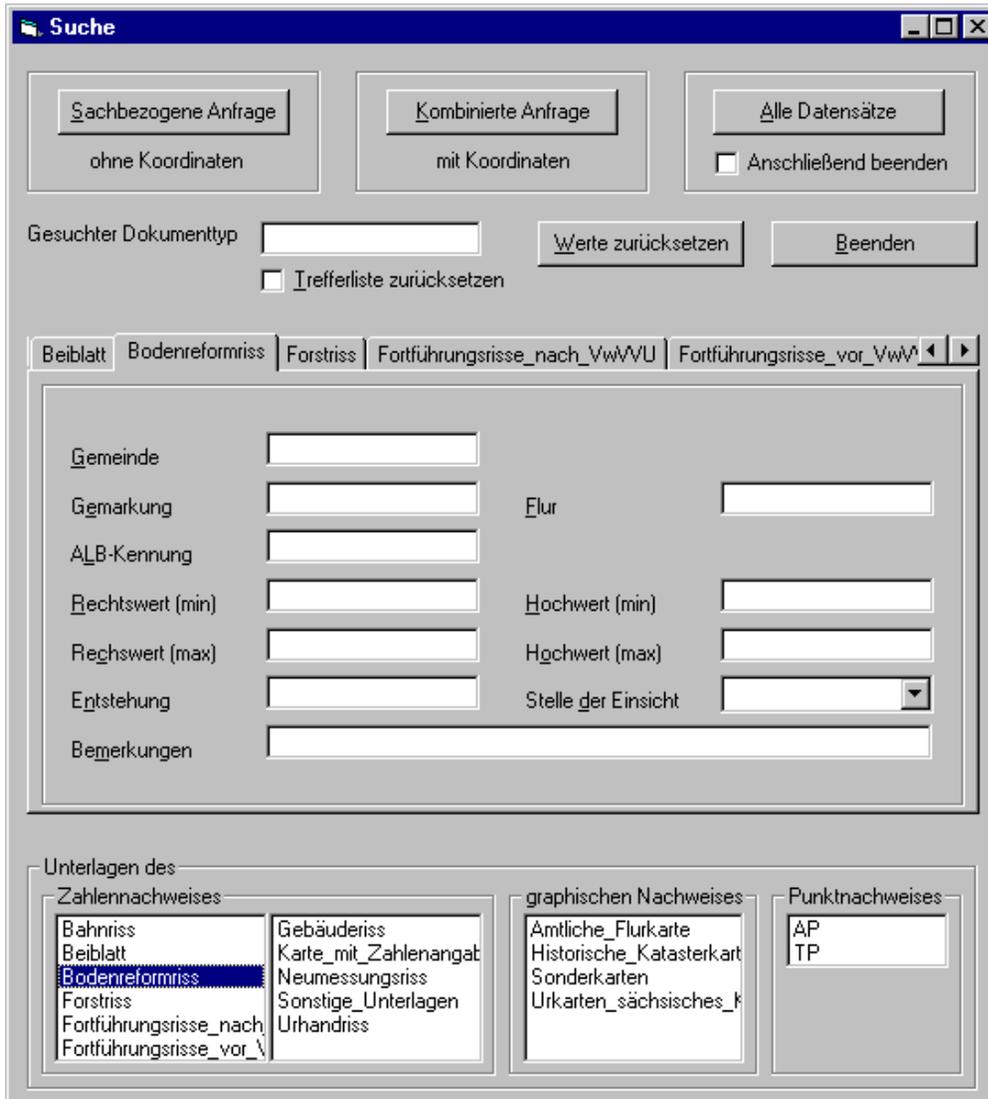


Je nach **Dokumententyp** sind unterschiedliche **Sachdaten** definiert, die beim Einrichten eines Dokumentes anzugeben sind.

Die **Formulare** können in Form von VB-Applikationen nach eigenen Bedürfnissen erstellt werden oder es wird ein Formular dynamisch anhand der Sachdatendefinition generiert.

Beispiel für kunden-
spezifisches Formular

GDM – Suchen der Sachdaten per Suchmaske



The screenshot shows a search mask application window titled "Suche". It features several sections:

- Buttons:** "Sachbezogene Anfrage" (ohne Koordinaten), "Kombinierte Anfrage" (mit Koordinaten), "Alle Datensätze", "Anschließend beenden" (checkbox), "Gesuchter Dokumenttyp" (input field), "Werte zurücksetzen", "Beenden", "Trefferliste zurücksetzen" (checkbox).
- Navigation:** A set of tabs: "Beiblatt", "Bodenreformriss", "Forstriss", "Fortführungsrisse_nach_VwVWU", "Fortführungsrisse_vor_VwVWU".
- Search Fields:**
 - Gemeinde (input field)
 - Gemarkung (input field)
 - ALB-Kennung (input field)
 - Rechtswert (min) (input field)
 - Rechtswert (max) (input field)
 - Entstehung (input field)
 - Bemerkungen (input field)
 - Flur (input field)
 - Hochwert (min) (input field)
 - Hochwert (max) (input field)
 - Stelle der Einsicht (dropdown menu)
- Unterlagen des:**
 - Zahlennachweises:** Bahnriß, Beiblatt, **Bodenreformriss** (highlighted), Forstriss, Fortführungsrisse_nach_VwVWU, Fortführungsrisse_vor_VwVWU.
 - graphischen Nachweises:** Amtliche_Flurkarte, Historische_Katasterkarte, Sonderkarten, Urkarten_sächsisches_K...
 - Punktnachweises:** AP, TP.

Suchmasken können individuell als VB-Applikationen erstellt werden.

Ist keine Suchmaske definiert, dann wird eine Standard-suchmaske verwendet.

Beispiel für kunden-spezifische Suchmaske

GDM – Georeferenzierung von Dokumenten

The screenshot shows the GDM software interface. On the left, a file list contains various image files. The main window displays a topographic map with a document overlay. A 'Passpunkte' dialog box is open, showing a table of control points with their local and world coordinates and deviations.

Grafik x	Grafik y	Rechtswert	Hochwert	WGS84 Status
48	61	2610606.890	5565145.810	Abweichung: -10.583 / 57.820 m
2420	124	2611040.770	5565355.760	Abweichung: 11.185 / -61.107 m
2389	3255	2611917.490	5564043.100	Abweichung: -12.306 / 67.233 m
235	3358	2610760.860	5564031.090	Abweichung: 11.704 / -63.946 m

Additional text in the dialog box includes 'ungültige Koordinaten' and buttons for 'Zelle leeren', 'Zelle entfernen', 'Zelle hinzufügen', 'Speichern...', and 'Laden...'. The status bar at the bottom shows 'Umgebende Box: l/w= (2610569.851/5563904.470), r/o= (2611952.469/5565343.223) B-Fehler: 63.663 m'.

Die Georeferenzierung eines Dokumentes erfolgt über Passpunkte, d.h. über die Zuordnung von Dokumentkoordinaten zu Weltkoordinaten. Ist keine Geoinformation vorhanden, dann kann man mit Hilfe eines VB-Skriptes die Weltkoordinaten ermitteln lassen:

- anhand von Sachdaten,
- anhand des Dateinamens,
- durch Anfrage beim GDM-Server.

GDM – Unterstützung von Fremdformaten

RissClient - [rpc.tif]

Microsoft Excel - Bauplan2_24_98.xls

Beispiel für Excel-Datei, die per GeoReferenz ange...

		Zeitplan				Zeitplan				
		März	April	Mai	Juni					
1	Aufwand	10	12	14	16	18	20	22	24	26
2	1	m								
3	3	k	k	k						
4	4		m	m	m	m				
5	2					m	m			
6	1							m		
7	nn									
8	1		m							
9	vorh									
10	5									
11	1				k					

Drücken Sie F1, um Hilfe zu erhalten.

Verbindung	Karten-Koordinaten		Dokument-Id	Rec
	linkswert	Rechtswert		
links-unten	3381065.04	5557644.62	AVI_1_DEMO	339
rechts-oben	3408441.08	5594467.48	Bauplan_24/98	339
			Bauplan2_24/98	339
			RheinlandPfalz	339

Dokumente mit Fremdformaten werden mit Hilfe externer Programme (z.B. Viewer, MS-Office) angezeigt.

GDM – Administration vom GDM-Client

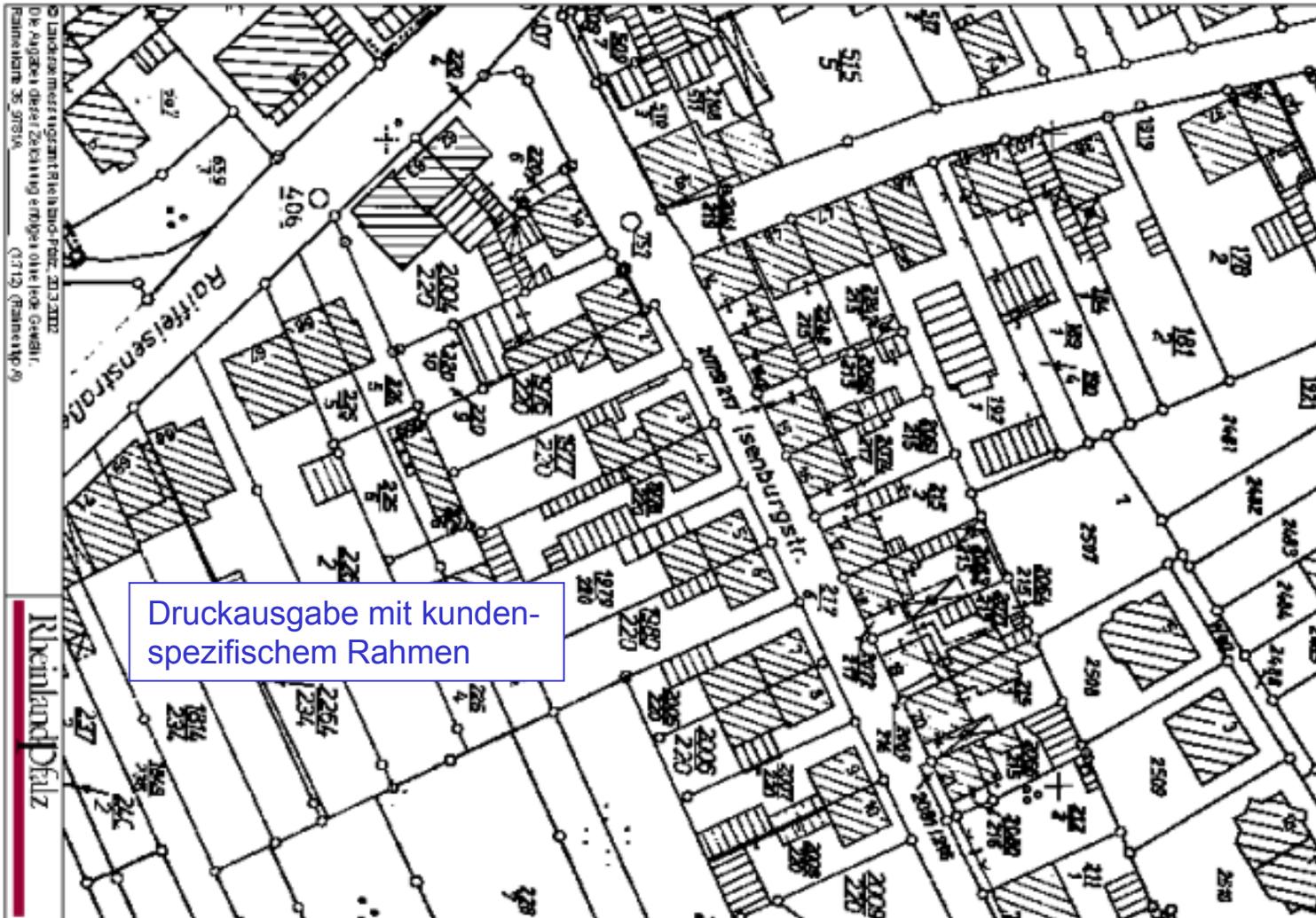
The screenshot shows a window titled 'Benutzer- und Server-Verwaltung' with tabs for 'Clients', 'Benutzer', 'Server-Kommando', and 'Zugriff DokTypen'. The 'Benutzer' tab is active, displaying a list of users on the left and a form for editing the selected user 'demo' on the right. The list includes 'loibl', 'buss', 'gast', and 'demo'. The form fields include: Benutzer-ID/Name (4), Name, Postadresse, Telefon, Fax, e-Mail, Zeit (Summe) (00:00:04), Letztes Einloggen (17.07.2000 13:08:19), and Status / Rechte (AuskunftPlatz / auskunft). A password field shows 'Gültig bis 06.12.1999 00:00:00' and a red message '1 Fehleingabe'. At the bottom are buttons for '4 löschen', 'Neuer Benutzer', 'Disable demo', 'Beenden', 'Neues Passwort', and 'Status ändern'.

Um die Verwaltung von Benutzern und ihren Rechten und von Dokumenttypen zu vereinfachen, kann ein Administrator bestimmte Aufgaben zur Steuerung des GDM-Servers vom GDM-Client aus durchführen:

- Benutzerverwaltung,
- Aktivieren und Deaktivieren von Benutzern,
- Benutzer, die zu oft versucht haben, sich mit einem falschen Passwort einzuloggen oder die zu oft unberechtigt privilegierte Befehle verwendet haben und deshalb deaktiviert wurden, wieder zu aktivieren,
- GDM-Protokollbefehle ausführen,
- den Zugriff auf Dokumenttypen regeln,
- Logins für Wartungsarbeiten ab- und anschalten,
- ...

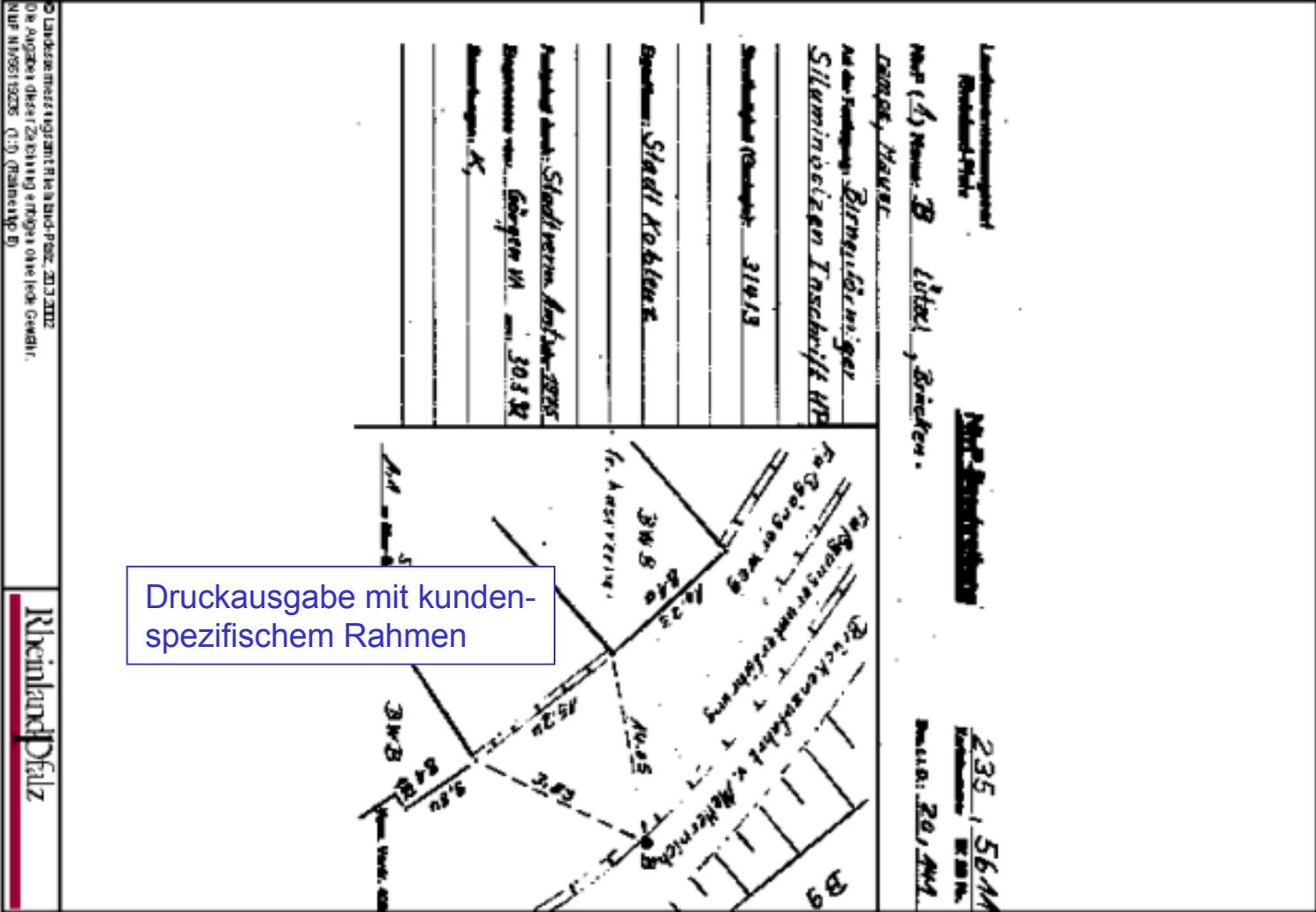
GDM – Datenausgabe am GDM-Client

GDM-Daten können gedruckt oder als Datei abgegeben werden:



GDM – Datenausgabe am GDM-Client

GDM-Daten können gedruckt oder als Datei abgegeben werden:

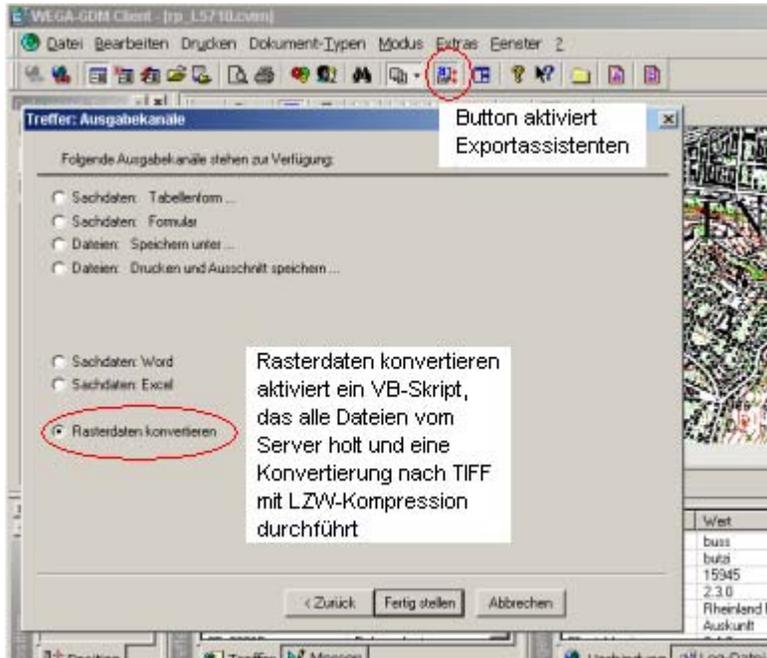


© Landesvermessungsamt Rheinland-Pfalz, 2013, 2012
 Die Angaben dieser Zeichnung erfolgen ohne jede Gewähr.
 Maf-Nr. 11/051/12026 (1:10, Rahmenmappe B)

Rheinland-Pfalz

Druckausgabe mit kundenspezifischem Rahmen

GDM – Datenabgabe am GDM-Client



Bei Abgabe der Daten in digitaler Form kann ein *Exportkanal* aktiviert werden, der die Dateien nicht nur vom GDM-Server holt, sondern z.B. auch nach TIFF konvertiert.

Anhand der vorliegenden Dokumentformate kann ein VB-Skript die Umsetzung der Dateien z.B. von TFW nach TIFF steuern. Dabei können in der Datenbank vorhandene Geoinformationen in eine TIFF-Datei eingebracht werden.

Die notwendigen *Konverterprogramme* werden beim Login vom GDM-Server zum GDM-Client übertragen (sofern auf dem GDM-Server neuere Versionen vorhanden sind). Es sind dies:

- **geotifcp** Umsetzung der Kompressionsart und Laden der Geoinformation,
- **layer2pal** Umsetzung von MultiLayer TIFF nach Palette-TIFF,
- **cvim2tiff** Umsetzung von CVIM nach Palette- oder RGB-TIFF,
- **tifftags** Ausgabe der TIFF-Tags wie Datenformat und Farbangaben.

GDM – Farbige Dokumente

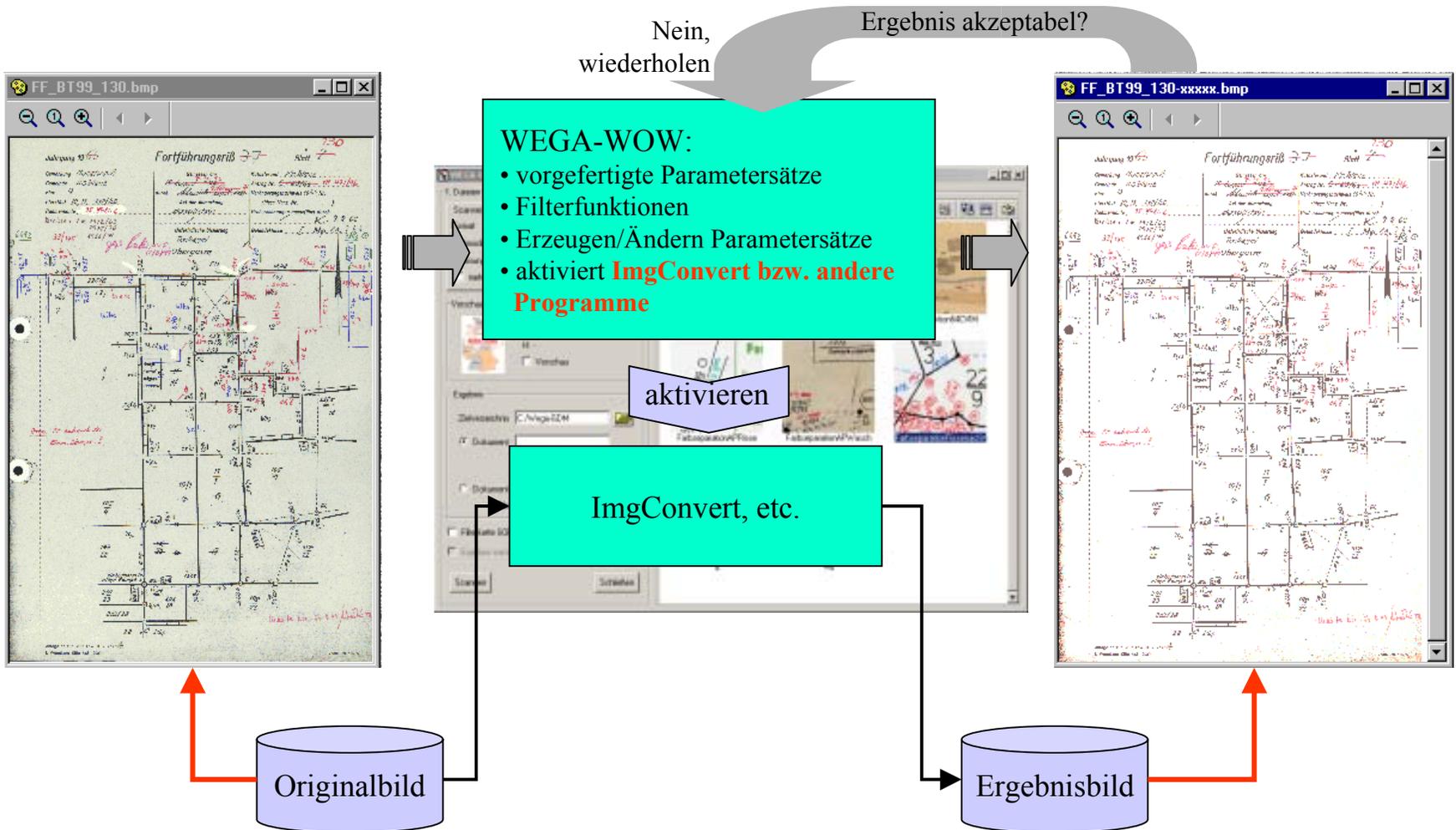
Intern darstellbare Farbbilder:

- Windows-BMP,
- CVIM (M.O.S.S. Bildformat),
- s/w-TIFF,
- 256-Farben-TIFF (=Palette-TIFF),
- farbseparierte Multi-Layer-TIFF,
- 24-bit-RGB-TIFF,
- JPEG,
- weitere Formate durch Anschluß der **FreeImage**-Grafikbibliothek.

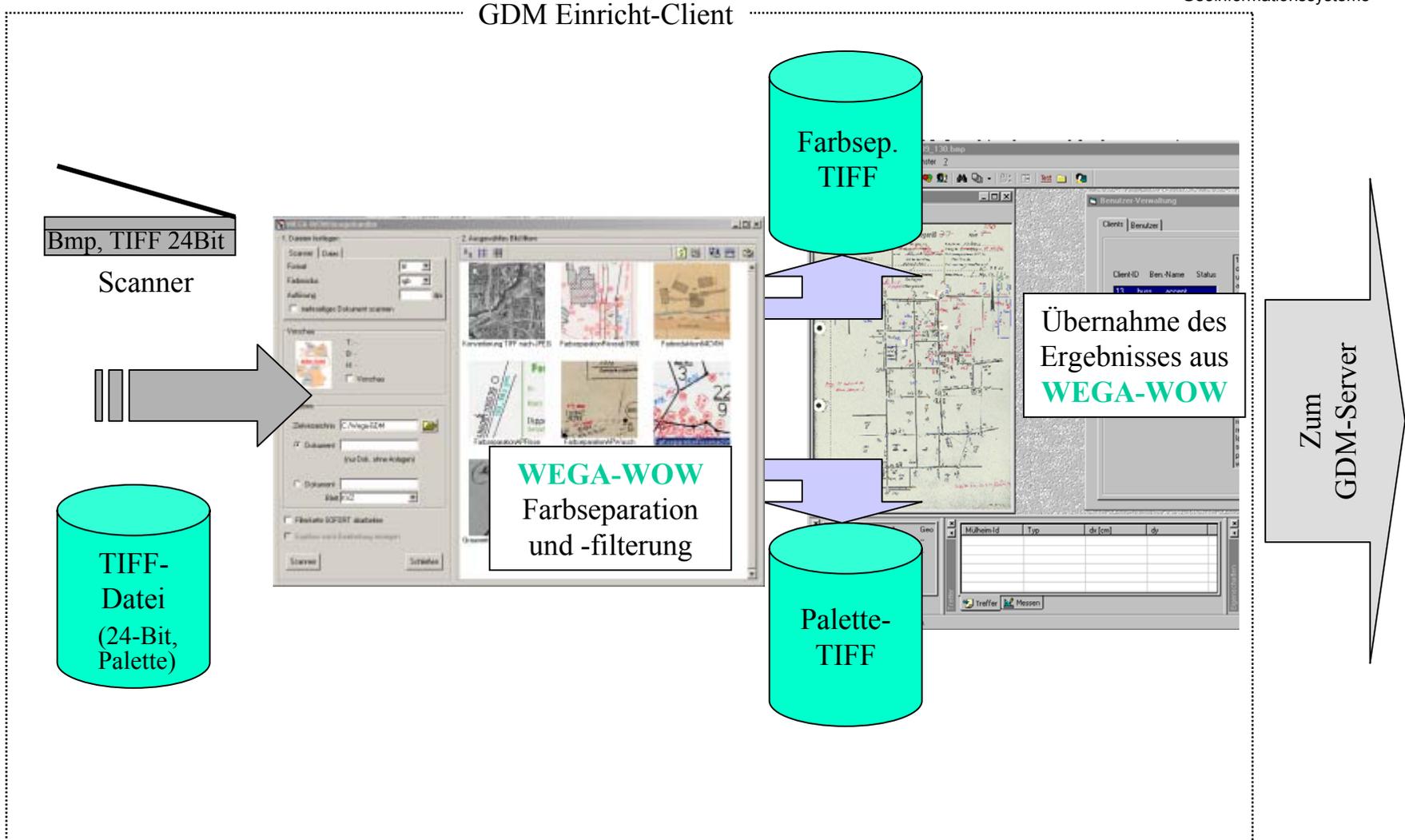
Die **großen Datenmengen** bei 24-bit-Farbbildern erfordern besondere Methoden:

- **Farbreduktion** von 24 Bit (ca. 16 Mio. Farben) auf 256 Farben, falls keine Farbseparation möglich.
- **Filterung** von Palette-TIFF-Bildern (8-Bit), um Datenmenge zu verringern.
- **Farbseparation** des Bildes in ca. 8-16 Farben (je nach Vorlage) und Speicherung der Farbebenen als Einzelbilder in TIFF-Datei und Ausgabe im GDM-Client durch Überlagerung der Farbebenen.

GDM – Datenreduktion durch Farbseparation

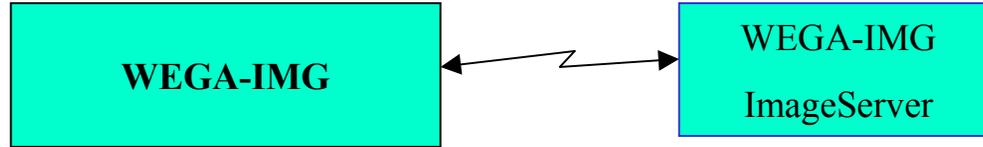


GDM – Farbige Dokumente einbringen



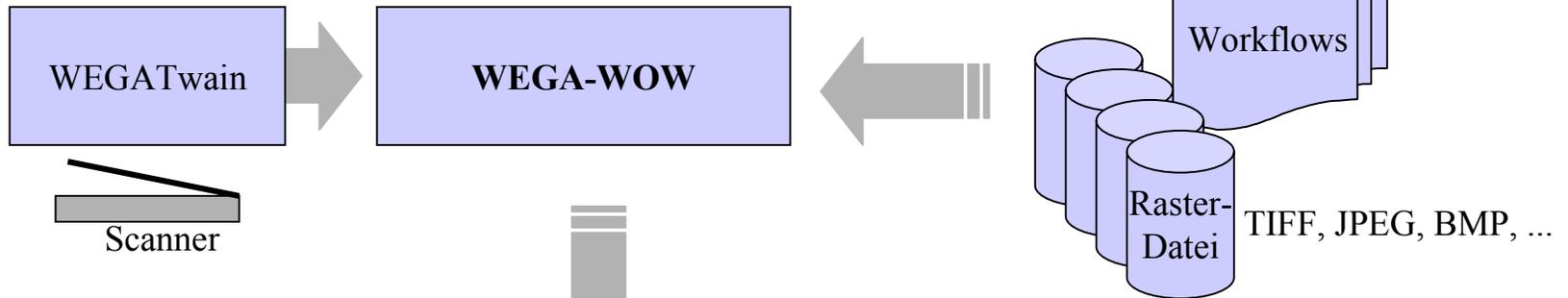
GDM – Einsatz der WEGA-Komponenten

Filterketten erstellen



Exportieren Filter-Workflow

Dokumente aufbereiten

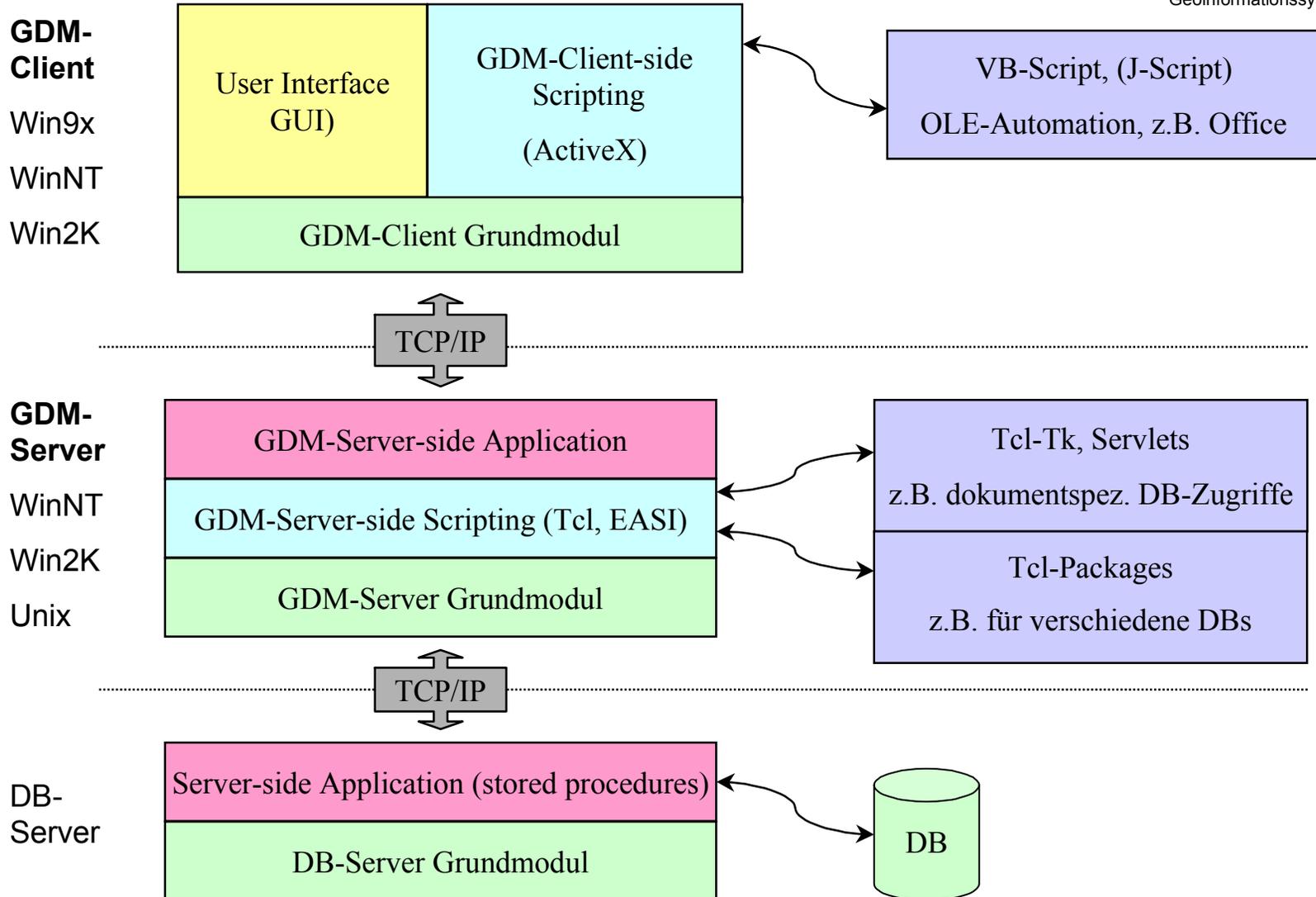


Dokumente einbringen,
suchen und verwalten

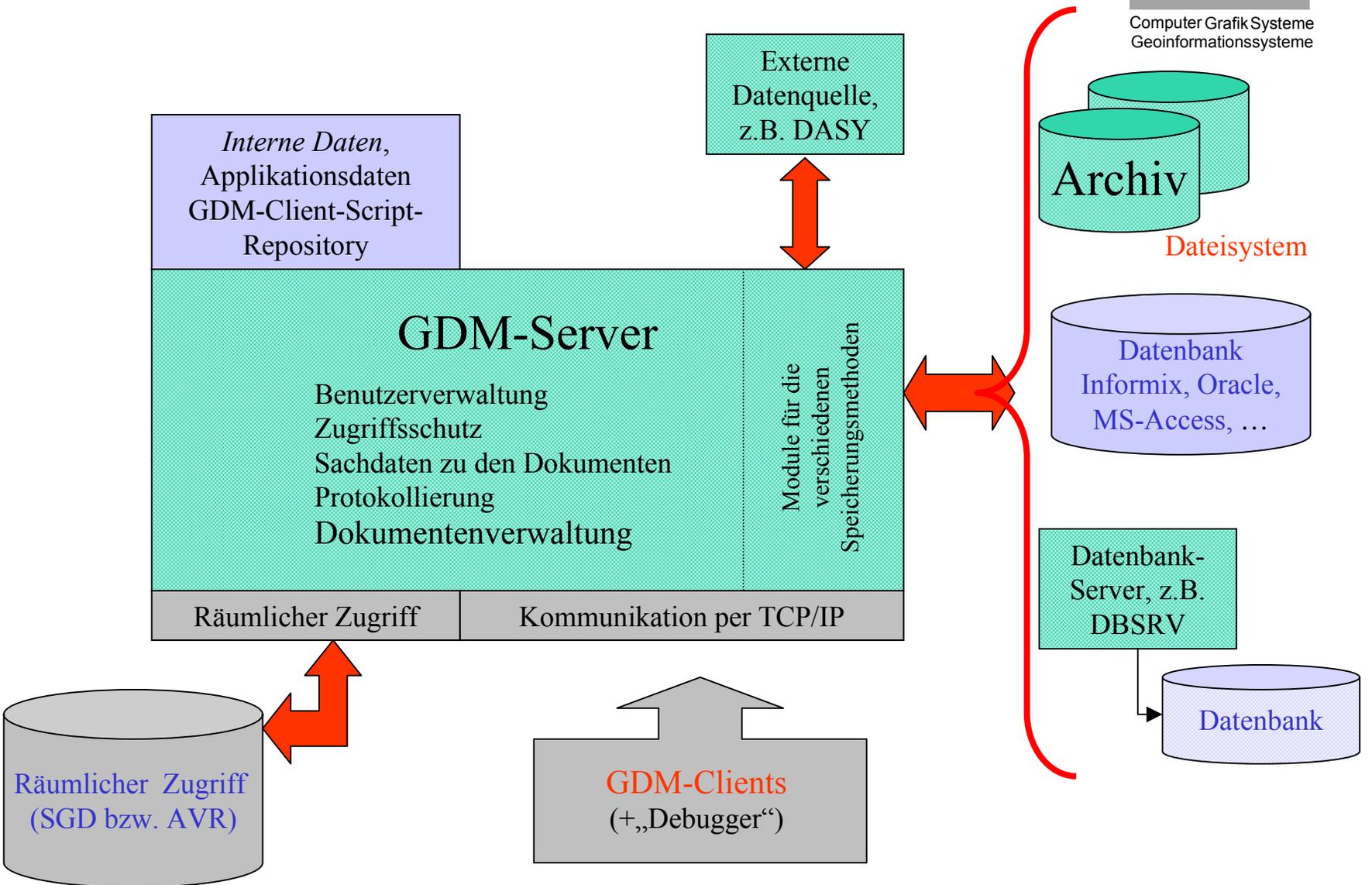
Aufbereitetes Rasterbild



GDM – Client/Server Technologien



GDM – Struktur GDM-Server



GDM – Realisierung (1)

GDM-Server

- kann unter UNIX (HP-UX, IRIX, Solaris, Linux) und Windows™ (WinNT, Win2k, WinXP) eingesetzt werden,
 - wurde in *Tcl* realisiert,
 - für den räumlichen Zugriff wird *EASI* benutzt (deshalb muß mindestens WinNT eingesetzt werden),
 - für den Datenbankzugriff werden die *Packages*
 - *IsqItcl* für Informix,
 - *Oratcl* für Oracle,
 - *tclodbc* für ODBC (unter Windows)
- eingesetzt,
- *spezielles Package* für den Server „*EcmDoRis*“ (z.B. für Verschlüsselung und Datumsangaben).

GDM – Realisierung (2)

GDM-Client

- kann nur auf Windows™-Plattformen eingesetzt werden,
- wurde in **C++** realisiert,
- für Oberfläche werden **MFC** (Microsoft Foundation Classes) eingesetzt,
- zur Entwicklung von clientseitigen Applikationen steht eine **ActiveX**-Schnittstelle bereit,
- bei bestimmten GDM-Clientfunktionen wird geprüft, ob **VB-Skripte** vorhanden sind, wenn ja werden diese automatisch ausgeführt,
- alle Skripte, Programme, VB-Applikationen werden vom GDM-Server beim Login geladen, sofern der GDM-Server aktuellere Versionen besitzt (automatisches Applikationsupdate).

GDM – Server-Verzeichnisstruktur (1)

- <server>** Enthält den GDM-Server mit alle notwendigen shared libraries und Skripten, besonders *ecmdoris.dll*, *ecmrun.dll* und *ecmzvfs.dll* (Extension richtet sich nach dem Betriebssystem).
- actlib** Enthält alle Aktionsroutinen für die GDM-Serverbefehle.
- lib** Enthält allgemeine Routinen (*bas_* → Basisroutinen, *clg_* → Geometrieroutinen, *dba_* → DB-Zugriff, *dok_* → Dokumenthandling, *geo_* → GDM-Serverroutinen. Die Prozeduren stehen in der BTL-Bibliothek (Binary Tcl Library).
- sql** SQL-Anweisungen, um eine DB aufzusetzen.
 - base** Alle Skripte mit SQL-Anweisungen und im GDM-Server erforderliche Daten.
 - <name>** Skripte mit SQL-Anweisungen für Applikationsdaten.
 - sachsen** *Sachsen*-GDM-Server mit Dokumenttypen und Sachdatendefinitionen.
 - ...
- srvfiles** Applikationsspezifische Daten und Skripte, unterteilt nach Art des Servers: Pfade, Lizenzinformation, spezielle Prozeduren zur Generierung der DokID.
 - <name>** Skripte und Daten für angegebene Applikation.
 - sachsen** Alle Dateien für *Sachsen*-GDM-Server.
 - ...

Alle **roten** Verzeichnisse werden ausgeliefert, die **blauen** sind erforderlich bei eigenen Entwicklungen an Servererweiterungen.

GDM – Server-Verzeichnisstruktur (2)

<server cont'd>

- sgd** Hier steht die SGD-Datei für die georeferenzierte Suche und die Log-datei mit allen Aktionen des Servers.
- files** Hier stehen die Dokumente, wenn Speicherung im Dateisystem vorgegeben wurde. Der Ort, an dem die Dokumente gespeichert werden, kann in der Konfigurationsdatei definiert werden.
- doc** Dokumentation des Servers, falls bei Installation ausgewählt.
- clntscripts** Verzeichnis, in dem alle Programme, DLLs, Skripte und sonstige Dateien gespeichert sind, die der GDM-Client für diese Installation benötigt. Der Bereich kann in weitere Unterverzeichnisse strukturiert werden, um verschiedene Serverinstanzen „fahren“ zu können. Beim Login-Vorgang werden diese Dateien mit den aktuellen Clientdateien verglichen und, falls unterschiedlich, erfolgt ein Download.
- clntmodules** Wenn vorhanden, enthält dieses Verzeichnis alle Module, Programme, Skripte und sonstige Dateien, die zur Aktualisierung der GDM-Client-Installation nötig sind. Beim Login-Vorgang prüft der GDM-Client, ob ein Update der GDM-Client-Installation nötig ist.

Alle rot-markierten Verzeichnisse werden ausgeliefert, alle grün markierten sind optional

GDM – Konfigurationsdatei „...srv.dat“ (1)

In dem Verzeichnis **<server> / srvfiles / <name>**

steht die Konfigurationsdatei mit den Informationen über die zu verwendende DB, die Lizenzierung, die Pfade zu den Serverdateien und -dokumenten, ...

Die Datei ist eine Tcl-Datei, jedoch mit der Extension „.dat“. Üblicherweise enthält der Dateiname den Text „srv“, z.B. „rpsrv.dat, sacsrv.dat“.

In dieser Datei sind in den Komponenten `serverID`, `licenseOwner`, `licenseCode`, `licenseFeat` die Identifikation des GDM-Servers und die Lizenzinformationen enthalten. Wobei die Komponente `licenseFeat` eine Liste mit Spezialfunktionen wie LZW-Kompression, Paßpunktdialog, GeoTIFF-Unterstützung, Annotation definiert. Je nach Lizenz sind diese Features hier aufgeführt und aktivieren diese Funktionalitäten im GDM-Server:

```

set geoSetup(serverID)           111
set geoSetup(applName)           democd
set geoSetup(licenseOwner)      "Demo CD"
set geoSetup(licenseCode)       AQ2PLSSYINC70S-CK%MF.WT
set geoSetup(licenseFeat)       "1 2 4 6"
set geoSetup(xfrReadBlockSize)   8192    ;# Blockgröße bei Dateilesevorgängen
set geoSetup(xfrWriteBlockSize)  4096    ;# Blockgröße Dateischreibvorgänge
  
```

GDM – Konfigurationsdatei „...srv.dat“ (2)

In der Konfigurationsdatei können auch applikationsspezifische Tcl-Skripte beim Start des GDM-Servers geladen werden (Komponente `srvSources`). Die hier angegebenen Prozeduren werden nicht per Bibliothekslademechanismus in den GDM-Server geladen, sondern stehen von Anfang an im Speicher des GDM-Servers.

```
set geoSetup(file2ppGenerator)    dasy_marTif2PP
set geoSetup(dokidGenerator)     dasy_marDokID
set geoSetup(srvSources)       "geomar mhdokdb mhgeschbuch,,
```

Es wird hier auch definiert, mit welcher Datenbank der GDM-Server arbeitet. Der GDM-Server arbeitet im „autocommit“-Modus.

```
set geoSetup(dbConnect)        {ODBC DemoCDDDB}
set geoSetup(dbDatetimeFmt)     "%d-%b-%Y %H:%M:%S,,
```

Die Informationen, wo im Dateisystem die Dateien zu suchen sind, wird hier ebenfalls festgelegt (bzw. Festlegung erfolgt mit der Installation).

```
set geoSetup(gdmPath)          [geoCheck4EnvVar WEGA_GDMPATH $pf 1]
set geoSetup(progPath)         [file join $geoSetup(gdmPath) server]
set geoSetup(sgdFilename)     $geoSetup(gdmPath) /sgd/democd.sgd
set geoSetup(filePath)        $geoSetup(gdmPath) /Files/democd
```

GDM – Konfigurationsdatei „...srv.dat“ (3)

Generieren Lizenzcode

Konfiguration: Rheinland-Pfalz

Software: WEGA-GDM

ServerID: 5

Lizenznehmer: LVermGeo Rheinland-Pfalz

Max. Anzahl Clients: 10

Features:

- GeoTIFF
- CPTdialog
- TWAIN
- Attachments
- Annotation
- LZWCompression
- LassoObjects
- ConnectedObjects

Lizenz Code: AQ2PLSXB2S2Z2H-K5%PW.W4

Featurelist: 1 2 4 6

Erzeugen Schließen

Die Lizenzierungskodes werden mit Hilfe des nebenstehenden Tools generiert und die Daten per Cut&Paste in die Konfigurationsdatei eingetragen.

Dieses Tool kann auch Kunden, die eine Landeslizenz besitzen, zur Verfügung gestellt werden.

- Der GDM-Server wird per *ServerID* identifiziert. Diese muss im Netz eindeutig sein.
 - Damit werden die Datenbanktabellen der Serverinstanz zugeordnet.
- Alle relevanten Tabellen sind in einem Dictionary hinterlegt.
 - Im GDM-Server erfolgt der Zugriff auf Tabellen per *tableID*.
 - Serverinterne Tabellen haben *tableID* ≤ 100 .
- Nicht nur die Namen der Tabellen, auch Spaltennamen und -typen sind hinterlegt.
 - Einfache Zuordnung von GDM-Typen auf spezifische DB-Datentypen.
 - Möglichkeit, zusätzliche Informationen zu hinterlegen, z.B. Spaltennamen in der DB und auf dem Formular.
 - Möglichkeit, eine *virtuelle* Tabelle festzulegen, d.h. die Daten werden per Prozedur geladen bzw. manipuliert (Zugriff auf externe DBs, berechnete Daten, ...).

GDM – DB-Struktur (2)

DoSysDataT

ServerID	Name	...	tableID
<i>n</i>	BeispielServer		

tblWegaGDM

tableID	tableName	...
0	Dictionary oder tblWegaGDM, wenn nur ein GDM- Server	

Dictionary oder tblWegaGDM

tableID	tableName	Descr
<i>id</i>	<i>Name</i>	...

Sachdatentabelle „Name“

col1	col2	...

tblWegaColumns

tableID	columnID	colName	GDM- TypID
<i>Id</i>	<i>cid</i>	<i>dbColName</i>	<i>tid</i>

tblWegaTypes

GDM- TypID	DBType- Name	GDMTyp- Name
1	CHAR	GDMSTRING

GDM – DB-Datentypen

GDM-eigene *Datentypen*, um unabhängig von DB-Datentypen zu sein. In einer eigenen DB-Tabelle **tblWegaTypes** sind die Datentypen hinterlegt, sowie deren Umsetzung für aktuell verwendetes DB-System.

Da die Sachdatentabellen im Dictionary beschrieben sind, kann man hier auch „computed types“ definieren, die dann bei der Speicherung in die DB, bzw. beim Auslesen aus der DB, entsprechend umgesetzt werden (z.B. der Typ `GDMTRANSFORM`).

- **GDMSTRING** Zeichenkette, evtl. bestimmter Länge
- **GDMSHORT** 6-Bit Integer
- **GDMLONG** 32-Bit Integer
- **GDMFLOAT** Kommazahl mit einfacher Genauigkeit
- **GDMDOUBLE** Kommazahl mit doppelter Genauigkeit
- **GDMBOOLEAN** Ja/Nein Feld
- **GDMDATE** Datum 1583...2199 als Tageszahl
- **GDMDATETIME** Datum und Uhrzeit als Tageszahl+Sekunden
- **GDMENUM** referenziert Eintrag in Katalog
- **GDMTRANSFORM** transformierte 5 Ziffern/Zeichen

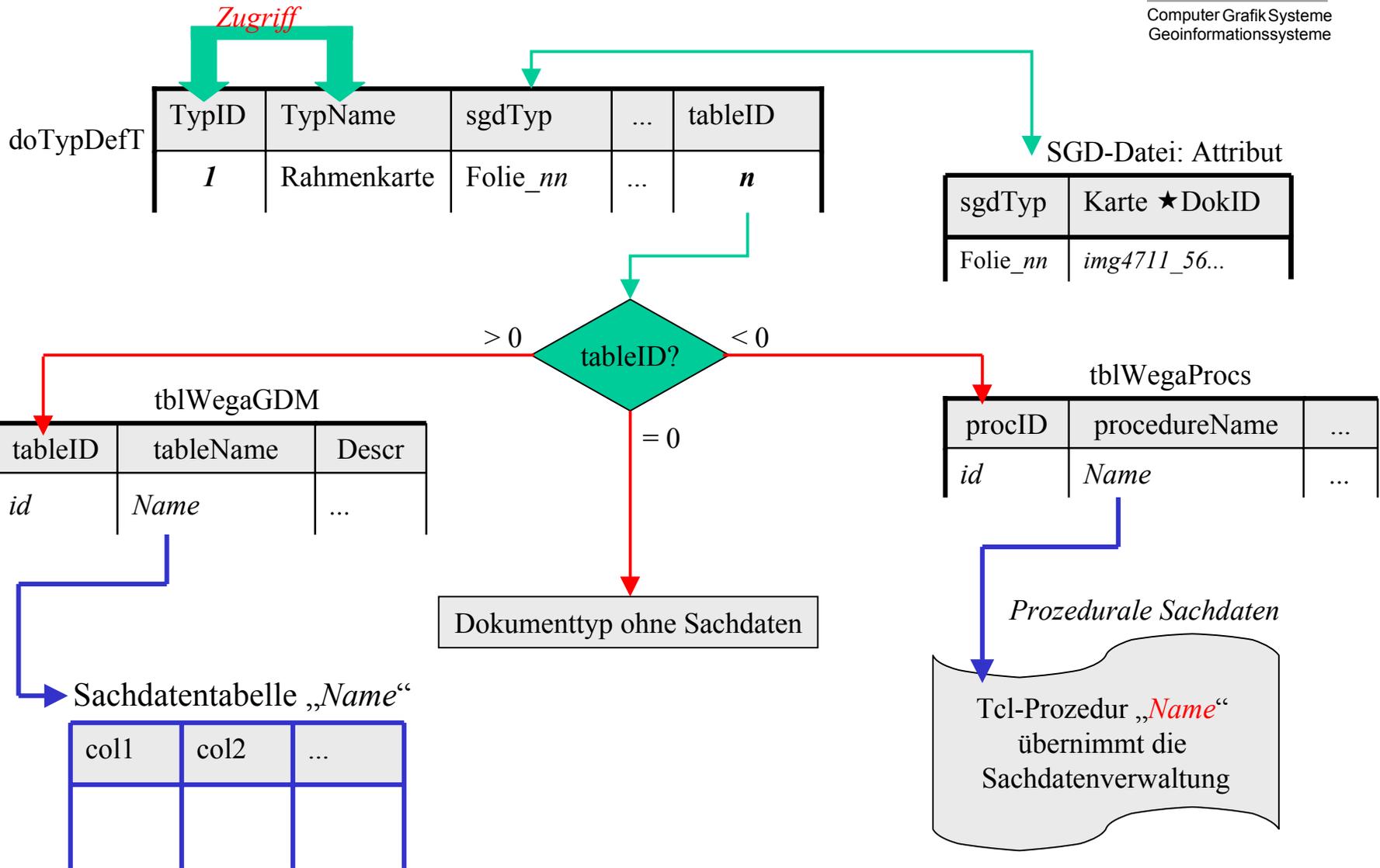
Datum- und Zeitangaben werden in einem GDM-eigenen Typ dargestellt. Dazu gibt es den Befehl „**ecmdoris datetime** ...“. Der Referenztag ist der 1.1.1900 und kann auch negativ sein (Angabe vor 1900). Die Angabe entspricht dem MS-VB-Script- und MS-Access-Datentyp `DateTime` (Double: ganzzahliger Teil → Tageszahl; Nachkommaanteil → Sekunden am Tag).

Man kann verschiedene Arten von Dokumenten in GDM verwenden: *Dokumenttypen*

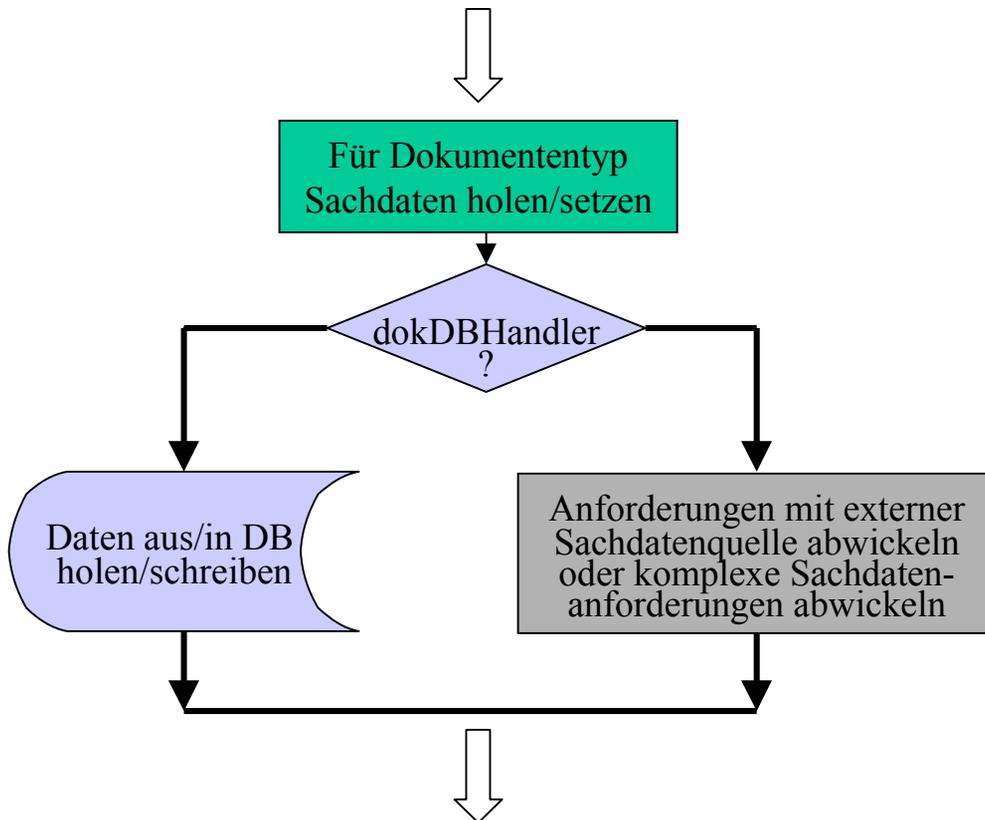
Definition der *Eigenschaften* des Dokumenttyps in der Datenbank (→ **doTypDefT**):

- mit Sachdaten (eine Tabelle/View),
- ohne Sachdaten (nur das Dokument an sich),
- mit komplexen Sachdaten (Daten in mehreren Tabellen, Daten aus externer Quelle/DB, ...)
- nur Sachdaten, jedoch Dokument,
- ob schreibgeschützt,
- Art der Speicherung, in Dateisystem, in DB-Blobs, per externer Dateiquelle, ...
- Dokument mit oder ohne Georeferenz,
- Dokument kann eine Karte sein, die dann mit anderen auch zu größeren Karten montiert werden können (z.B. Kacheln einer ALK). Diese Karten können wiederum zur geometrischen Abfrage von Dokumenten verwendet werden.
- Dokument ist Mappe, d.h. es verweist auf andere Dokumente (hierarchische Struktur) oder darf in eine Mappe eingetragen werden.
- Zugriffseinschränkungen auf bestimmte Benutzer,
- Art der Dokumentenkennung, z.B. abgeleitet aus Sachdaten, automatisch generiert.

GDM – Sachdaten je Dokumenttyp



GDM – Externe/komplexe Sachdaten



Manchmal möchte man *Sachdaten* zu einem Dokument holen, die nicht in der gleichen Datenbank enthalten sind, in der der GDM-Server seine Daten hält, z.B. Zugriff auf das *ALB*, das auf einem anderen Rechner und/oder in einer anderen DB gehalten wird. Andererseits können die Sachdaten so komplex sein, daß die Standardfunktionalität des GDM-Servers nicht ausreicht.

Es wird daher immer, wenn der GDM-Client bzw. eine Applikation Sachdaten zu einem Dokument benötigt bzw. Sachdaten einbringen will, bei jedem Dokumententyp geprüft, ob für diesen Typ ein sogenannter **dokDBHandler** definiert ist. Wenn ja, wird die Anfrage an diesen Handler weitergeleitet. Dieser Handler muß dann die Kommunikation mit der externen Sachdatenquelle abwickeln.

GDM – Neuen Dokumenttyp aufsetzen (1)

- Festlegen der Eigenschaften (*Konzeption*),
- Eintragen der Eigenschaften in Datenbank-Tabelle `doTypDefT` (online-Erweiterung),
- in SQL-Skript `<server>/sql/<app>/dotypdefvalues.tcl` neue Zeile einfügen:

```
set values {
  {1, 'Rahmenkarte', 'tif', ,Raka aus DASY', 'rahmenkarte', ..., ... }
  {3, 'doc', 'doc', 'WinWord', 'Folie_1', 35954.52, ..., ... }
  { ..., ..., ..., ..., ..., ... }
  {id, 'doktypeName', , 'fmt', 'Beschreibung', 'Folie_n', 36449.3812963, ..., ... }
}
```

- Wenn Dokument-Sachdaten verwendet werden, dann je nach *Sachdatenzugriffsform* Einträge in Datenbank und zugehörigen SQL-Skripten (bzw. neue SQL-Skripte),
- Sachdatentabellen anlegen „CREATE TABLE *name* ...“ → `<server>/sql/<app>/name.tcl`,
- in Tabelle `tblWegaGDM tableID` vergeben und neue Zeile(n) einfügen,
- SQL-Skript entsprechend `<server>/sql/<app>/tblwegagdmvalues.tcl` aktualisieren:

```
set values {
  {100, 'mhRissTable',      'Datentabelle Risse Mülheim a.d.Ruhr', '1', 36317 }
  {101, 'mhFlurModifTable', 'Tabelle der veränderten Flurstücke MH', '1', 36332 }
  { ..., ..., ... }
  { tabid, 'tableName', 'Beschreibung', '1', date}
}
```

GDM – Neuen Dokumenttyp aufsetzen (2)

- In Tabelle **tblWegaColumns** **tableID**, **columnID** vergeben und Spalteninformationen einfügen,
- SQL-Skript `<server>/sql/<app>/tblwegacolumnsvalues.tcl` aktualisieren, **tableID** und **Columns** eintragen,
- evtl. in Tabelle **tblWegaProcs** **tableID**, **procID** vergeben und Prozedurinformationen einfügen,
- dann SQL-Skript `<server>/sql/<app>/tblwegaprocsvalues.tcl` aktualisieren

```
set values {
  { -1, 'mh_DokDBhandler', 'Für Datentyp RissDatei DB-Daten-Funktionalität'}
  { -n, 'myOwnDokDBhandler', , 'Beschreibung....' }
}
```

- und angegebene Tcl-Prozedur **myOwnDokDBhandler** schreiben und in eigene Datei speichern,
- Vergabe der **Dokumentkennung (DokID)** überlegen und festlegen:
 - anhand des Dateinamens (default), z.B. topographischen Karten,
 - Generator anhand der Sachdaten, z.B. **geomar.tcl**,
 - Unique-ID-Generator.

Dementsprechend sind Prozeduren zu erstellen, die in der Konfigurationsdatei als zusätzliche Quellen geladen werden müssen.

GDM – Neuen Dokumenttyp aufsetzen (3)

- Falls der Dokumenttyp eine Karte ist, dann überlegen, wie die **Geoinformationen** eingebracht werden können:
 - anhand des Dateinamens, →Prozedur nötig, um Geoinfo abzuleiten,
 - anhand eines Paßpunkt-Generators →Prozedur nötig,
 - anhand Geoinformationen in der Datei, z.B. GeoTIFF.
- Zugriff auf Dokumente festlegen (**Dateiablage**):
 - Speicherung in Dateisystem, Festlegung Pfad, Suchfolge, ...
 - in BLOBs der Datenbank, asynchroner Transfer nötig,
 - per Handler, der Zugriff auf externe Quellen regelt: Handler schreiben.
- Eventuelle **VB-Skripte** für den GDM-Client erstellen bzw. anpassen, damit z.B. die Darstellung von Treffern oder der Export der Dokumente dieses Typ gesondert behandelt werden.
- Überlegen, ob **Patch-Skript** für existierende Installation erforderlich ist.

GDM – Datenbank aufsetzen (1)

Bevor man die Datenbank aufsetzen kann, muss man die *Dokumenttypen* festlegen:

- Welche *Sachdaten* sollen gespeichert werden → Tabellendefinition?
- Welche *Dokumentformate* sind erlaubt → TIFF, JPEG, BMP, ...?
- Sind die Dokumente *georeferenziert*?
 - Kann die Georeferenz abgeleitet werden, z.B. durch DB-Informationen oder aus dem Dokumentnamen (z.B. bei Kartennamen)?
- Sind spezielle *Features*, z.B. Annotationen oder Attachments, erlaubt?
- Darf das Dokument *geändert* oder *gelöscht* werden?
- Ist das Dokument eine *Karte*, so dass es zur Auswahl anderer Dokumente herangezogen werden kann?
- Wie ist das Dokument *gespeichert* oder zu speichern → Blob, Filesystem, externe Datenquelle, ...?

GDM – Datenbank aufsetzen (2)

Eine Datenbank für den GDM-Server wird erstellt durch Tcl-Skripte, die SQL-Anweisungen zum Erstellen und Laden von DB-Tabellen enthalten.

Die Skripte sind in dem Unterverzeichnis `<server>/sql/...` hinterlegt. Die Dateinamen unterliegen einer Namenskonvention:

`<tabellenname>.tcl` Erzeugen der DB-Tabelle(n), inklusive Index(e)

`<tabellenname>values.tcl` Laden der zuvor definierten Tabelle mit Daten

z.B. `dotypdef.tcl` erstellt die Tabelle `DoTypDefT` (Definition der Dokumenttypen) mit sqlCommand „CREATE ...“-Anweisungen, während `dotypdefvalues.tcl`, die eigentlichen sqlCommand „INSERT ...“-Anweisungen enthält, um die gerade erstellte Tabelle zu füllen.

Um die Skripte einsetzen zu können, muß man in einer Tcl-Shell zuallererst das Skript `dbinitialize.tcl` ablaufen lassen. Dann wird speziell der Befehl `sqlCommand` definiert, den die Skripte verwenden.

Man sollte vorher **genau auf die Einstellungen achten**, die in `dbinitialize.tcl` vorhanden sind (Art der DB, ob Löschen der Tabellen (!!)) oder nur Initialisierung, ...).

GDM – Datenbank aufsetzen (3)

Die *Initialisierung* läuft in 2 Schritten ab:

- Zuerst Definieren und Laden der GDM-Basistabellen (die Skripte hierfür sind im Verzeichnis `<server>/sql/base`)
- Danach Definieren und Laden eigener Tabellen und Erweitern der Basistabellen um Dokumenttypdefinitionen, um vordefinierte Benutzer, ... (Diese Skripte sind im applikationsspezifischen Verzeichnis `<server>/sql/<appl>` hinterlegt)

Das Erzeugen von Tabellen bzw. das Laden von Werten erfolgt somit folgendermaßen:

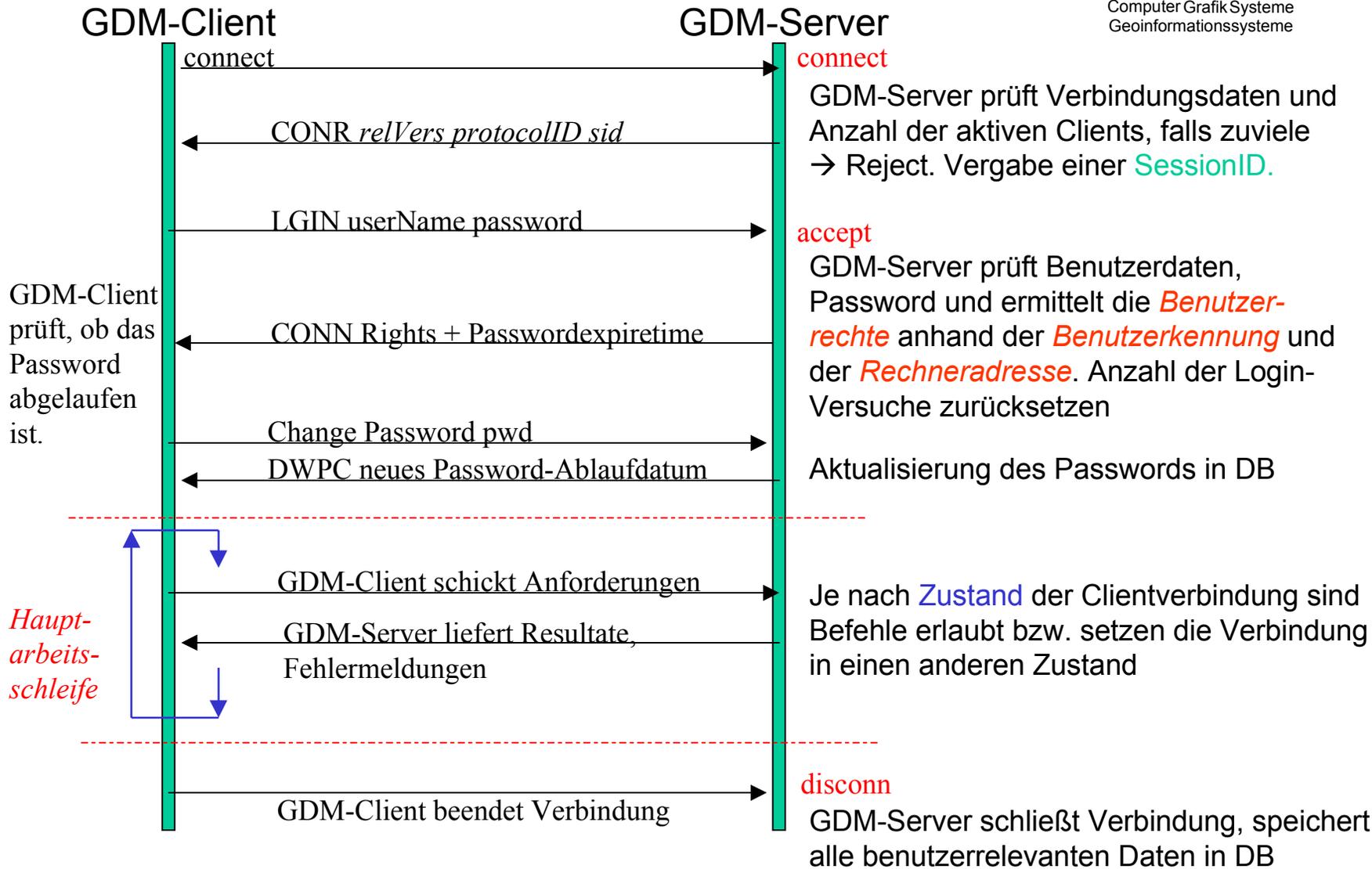
```
% tclsh
% source dbinitialize.tcl
% cd base oder cd ../<appl> #je nach Tabelle/Daten
% source dotypdeft.tcl
% source dotypdeftvalues.tcl
% sqlDisconnect
```

Man kann die DB-Initialisierung auch automatisch ablaufen lassen, sodass am Schluß die DB entsprechend aufgesetzt ist.

GDM – GDM-Server Startvorgang

- Der Startvorgang läuft in *Phasen* ab, in einigen Phasen kann man eingreifen (mit Hilfe von sogenannten *phased servlets*).
- Phasen:
 - 0 Laden der Konfigurationsdatei mit den serverspezifischen Daten.
 - 1 Initialisieren der internen Caches, Timer, Anlegen der Log-Datei, „Erlernen“ des Befehlssatzes (Kommunikationsprotokoll).
 - 2 Verbinden mit der DB, Öffnen und Initialisieren der SGD. Es werden die Tcl-Packages **EcmDoRIS**, **Ecmzvfs**, ... Geladen.
 - 3 Laden der Serverinformationen (mit Hilfe der **serverID** aus der srv-Datei), sowie der Systemtabellen, Dokumentdefinitionen und Prüfung der Lizenz.
 - 4 GDM-Server-Socket einrichten, Disconnect der SGD, bis ein Benutzer sich einloggt, und Warten auf Anforderungen von Clients.

GDM – Ablauf Login



GDM – DB-Zugriff für Benutzerrechte

Benutzer loggt sich ein mit Username

↓

doUsersT			
UserID	Username	password	aclID
<i>uid</i>	demo	(verschlüsseltes Password)	<i>aclID</i>

doUserRightsT		
UserID	rightsID	Add/Remove Flag
<i>uid</i>	<i>rid</i>	1 oder -1

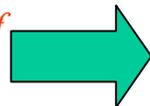
doAccListT	
aclID	Basisrecht + FeatureID
<i>aclID</i>	<i>facilID</i>

doRightsEnumT	
rightsID	rightsName
<i>rid</i>	Benutzerrecht

doFeaturesT	
facilID	Feature
<i>facilID</i>	FeatureName

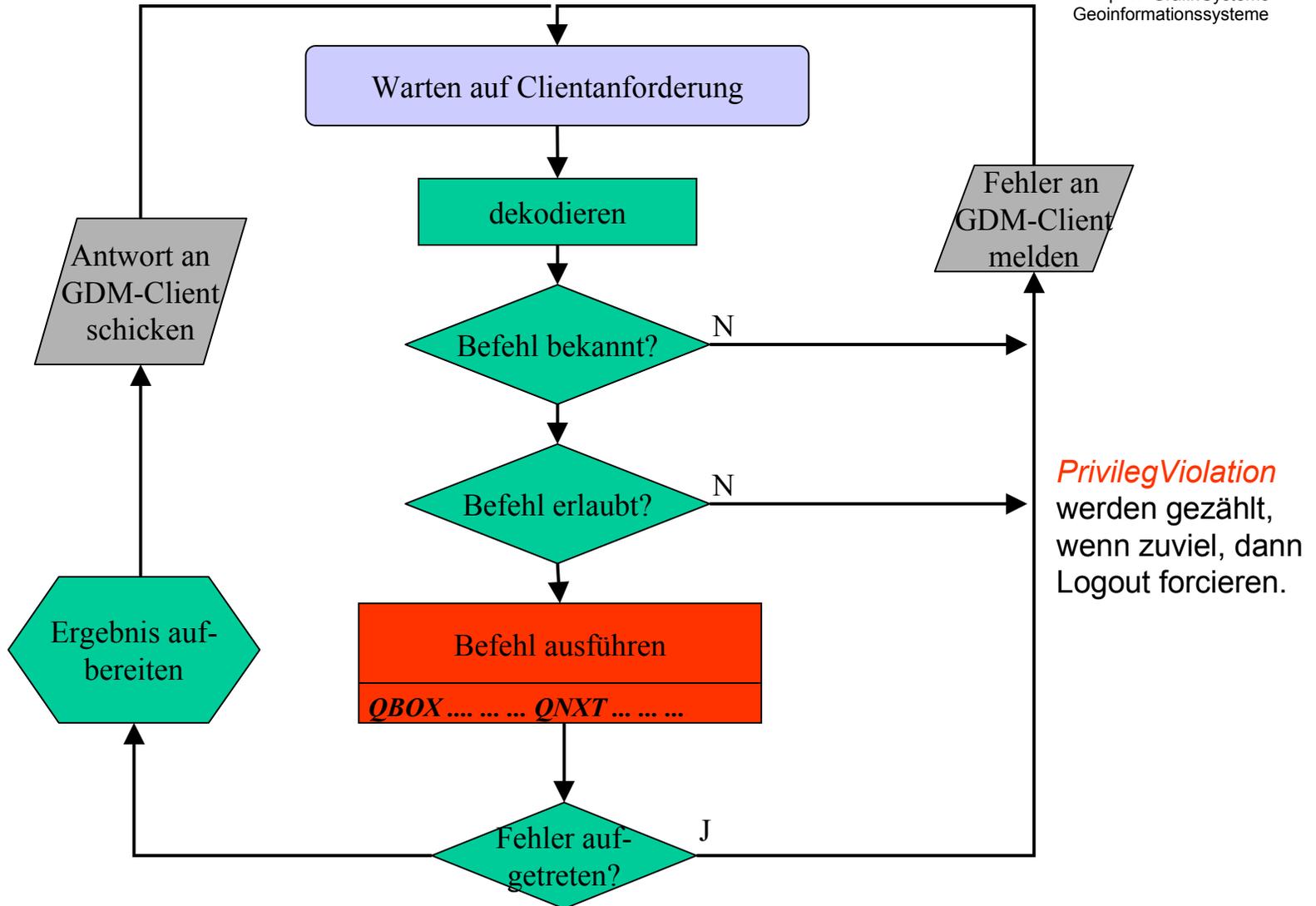
Aufbereitete Benutzerrechte

Benutzer befindet sich auf Rechner mit IP-Adresse



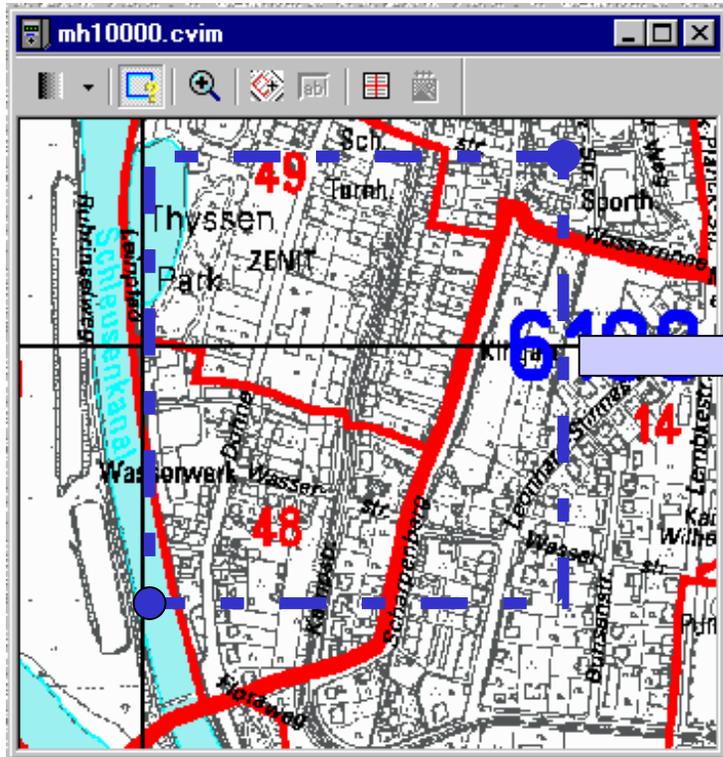
doTrustedHosts	
IP-Address Mask	aclID/rightsID
192.9.200.0	<i>aclID</i>

GDM – Befehlszyklus im GDM-Server



GDM – Ablauf einer Clientanforderung (1)

1. GDM-Client fordert Dokumente für einen bestimmten Bereich (Rechteck).



Suche alle Dokumente im Bereich
P(links unten) ... P(rechts oben)

GDM-Client Request:

QBOX {2561018 5698236 2561508 5698625}

2. GDM-Server führt Selektion mit Hilfe der räumlichen Zugriffsmethode aus:
Das Ergebnis ist eine **Collection** mit den gefundenen Treffern.

GDM – Ablauf einer Clientanforderung (2)

3. Der GDM-Server teilt dem GDM-Client mit, wieviele Treffer gefunden wurden:

```
XOBQ colhd2 2
```

und wartet danach auf die nächste Anforderung.

4. Der GDM-Client beginnt die Treffer der Reihe nach abzufragen (Query Next):

```
QNXN colhd2
```

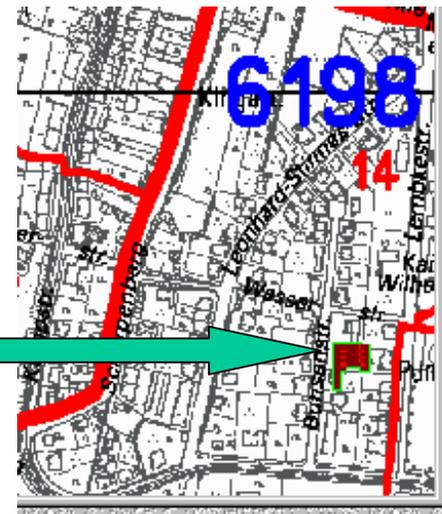
5. Der GDM-Server prüft, ob noch Treffer in der Collection sind, und liefert Ergebnis:

```
TXNQ colhd2 150 {2561500 5698250 Rahmenkarte 6198.0 {...} 6198_0.tif}
```

4. Der GDM-Client analysiert Antwort:

- zeichnet Symbol in der Übersichtskarte,
- aktualisiert die Liste der Treffer.

Mülheim-Id	Typ	dx [cm]	dy
6198.9	Rahmenkarte	?	?
6198.0	Rahmenkarte	?	?



GDM – Protokollierung der Aktionen

Diese Aktionen werden beim GDM-Client und beim GDM-Server protokolliert:

```
id=0x00b5 (12:02:45, 15255596): QueryBox: lu=(2561018.19, 5698236.77), ro=(2561508.85, 5698625.82),
docTypes='', loadDate='', docDate=''
id=0x00b5 (12:02:45, 15255626): <- QueryBox: strHandle='colhd12', count=2
id=0x00b5 (12:02:45, 15255636): QueryNext: handle='colhd12'
id=0x00b5 (12:02:45, 15255696): <- QueryNext: isn=150, posX=2561500.00, posY=5698250.00,
typ='Rahmenkarte', docId='6198.0', docName='6198_0.tif', 4 Koordinaten: (2561005.000 5698005.000)
(2561995.000 5698005.000) (2561995.000 5698495.000) (2561005.000 5698495.000)
id=0x00b5 (12:02:45, 15255726): QueryNext: handle='colhd12'
id=0x00b5 (12:02:45, 15255746): <- QueryNext: isn=151, posX=2561500.00, posY=5698750.00,
typ='Rahmenkarte', docId='6198.9', docName='6198_9.tif', 4 Koordinaten: (2561005.000 5698505.000)
(2561995.000 5698505.000) (2561995.000 5698995.000) (2561005.000 5698995.000)
id=0x00b5 (12:02:45, 15255766): QueryNext: handle='colhd12'
id=0x00b5 (12:02:45, 15255766): <- QueryNext: isn=0, posX=0.00, posY=0.00, typ='', docId='',
docName='', 0 Koordinaten:
id=0x00b5 (12:02:45, 15255766): QueryDone: handle='colhd12'
id=0x00b5 (12:02:45, 15255776): <- QueryDone
```

GDM-Client

```
00/02/28 11:06:27: Try to connect to sgdfilere...
00/02/28 11:06:27: Using SGD-File 'D:/Wega-GDM/sgd/mhpc.sgd'
00/02/28 11:06:27: SGD-File delete coll and context
00/02/28 11:06:27: Server goes to main loop
00/02/28 11:06:34: 014:sock216:1: file read cancelled
00/02/28 11:14:19: Incoming connect request sock200, 192.9.200.48, 1073
00/02/28 11:14:19: Server goes to main loop
```

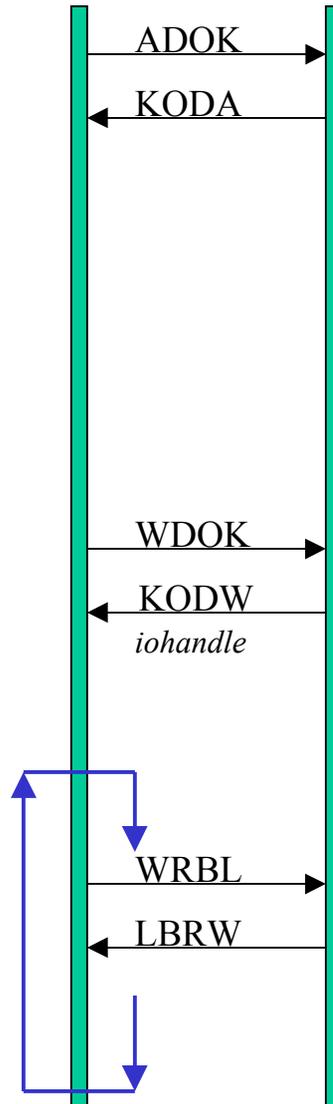
GDM-Server

GDM – Einrichten eines Dokumentes (1)

GDM-Client *meldet* an GDM-Server, dass er ein Dokument einbringen möchte.

GDM-Client hat *Freigabe* erhalten, und will nun das Dokument blockweise übertragen.

GDM-Client schickt die Daten *blockweise*



GDM-Server *genehmigt* dies: Checkt, ob Dokumenttyp zulässig, ob Dokument bereits vorhanden, oder bereits gelockt, d.h. ein anderer Benutzer bringt das gleiche Dokument ein. Es wird ein temporärer Lock auf dieses (noch nicht existierende) Dokument angelegt. Vorläufige Einträge in der DB und SGD-Datei (falls georeferenziert) und Eintrag in der zentralen Dokumentenquerverweistabelle *doGeoXTableT*.

GDM-Server bereitet den Datentransfer vor: Ermitteln, wo das Dokument zu speichern ist, (Prüfung auf freien Speicherplatz), Datei anlegen, I/O-Handle zur Referenzierung auf Datei allozieren und an GDM-Client liefern.

GDM-Server nimmt Daten in Empfang: Speichern der Daten in Datei; jeder Transfer wird vom GDM-Server bestätigt.

GDM – Einrichten eines Dokumentes (2)

Abspeichern der **DokID** in der Dokumentenquerweistabelle:

doGeoXTableT

DokID	DokTypeID	Datum	...	pntCnt	pntID
...	...			<i>n</i>	<i>pid</i>

doGeoXPntsT

pntID	lfdNr Punkt	x	y
...

Einrichtpolygon
X, Y, ...



Eintrag in
SGD-Datei

Einrichtpolygon als
Fläche mit Attribut
DokID, DokTyp

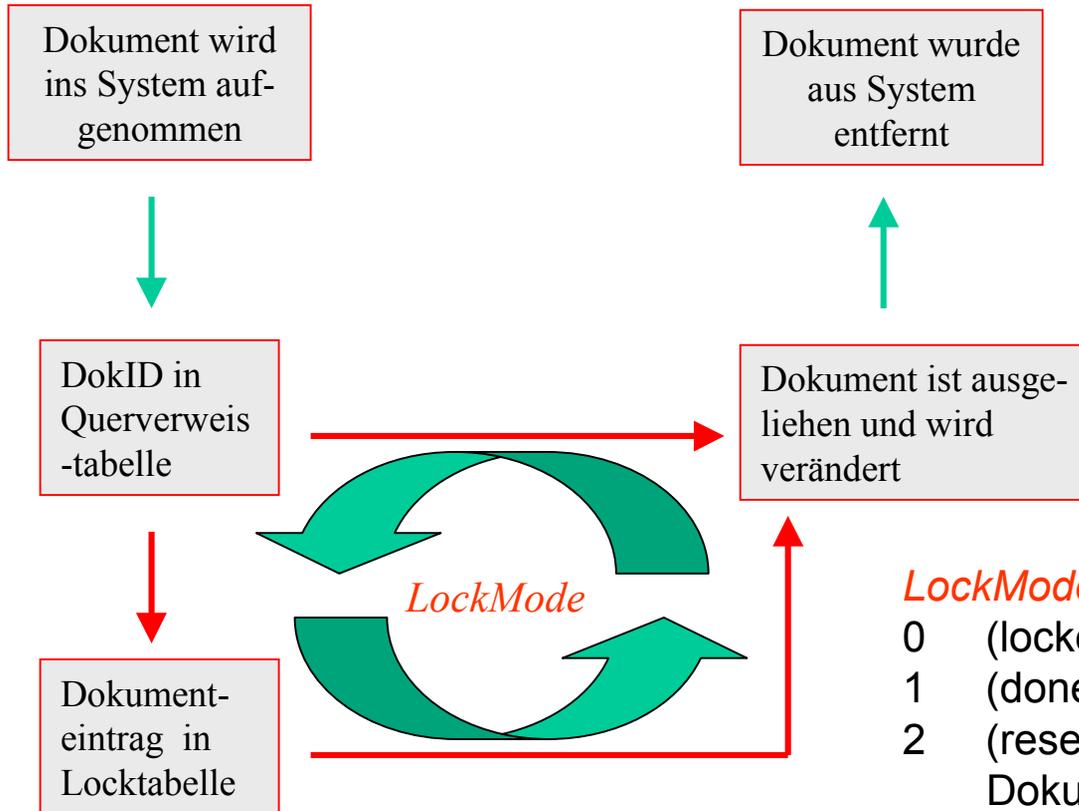
doGeoHotSpT

pntID	x	y
...

Referenzpunkt
hotSpotX, hotSpotY

falls Dokument mit Georeferenzierung

GDM – Zustände eines Dokumentes (1)

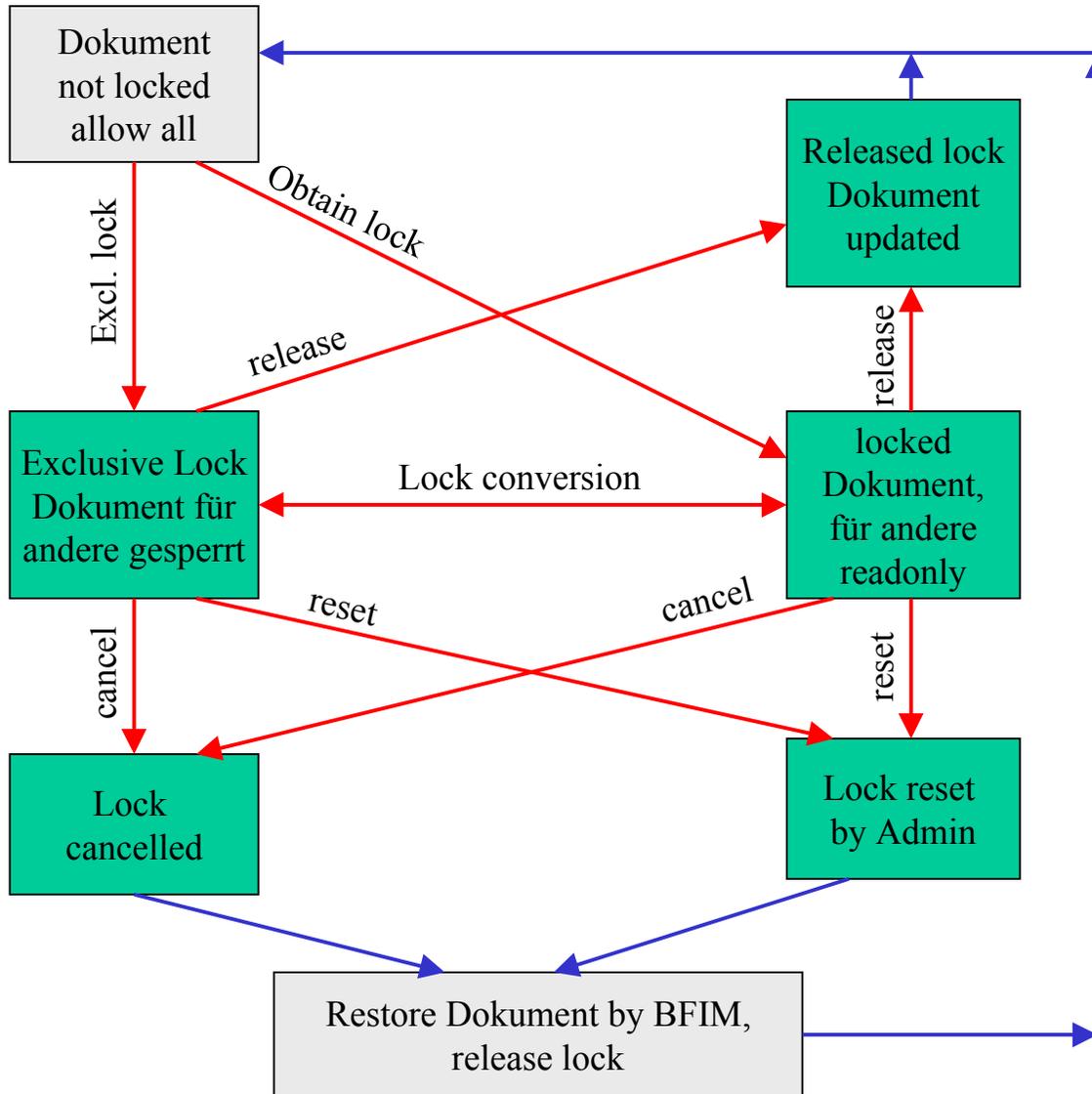


Alle Informationen über
 - Einbringen ins System,
 - Veränderungen,
 - Löschung
 sind zeitlich und bearbeitermäßig
 nachvollziehbar.

LockMode

- 0 (locked) in Bearbeitung
- 1 (done) Dokument wurde bearbeitet.
- 2 (reset) Bearbeitung des Dokumentes wurde angemeldet, aber von Administrator wieder zurückgesetzt.
- 3 (cancelled) Bearbeitung des Dokumentes wurde angemeldet, aber vom Bearbeiter abgebrochen.

GDM – Zustände eines Dokumentes (2)



Alle verändernden Transaktionen werden in der Datenbank mit Hilfe des Sperrmechanismus aufgezeichnet.

tblWegaLock

DokID	DokTyp	Lock-Mode	LockID
...	...	<i>zustand</i>	...

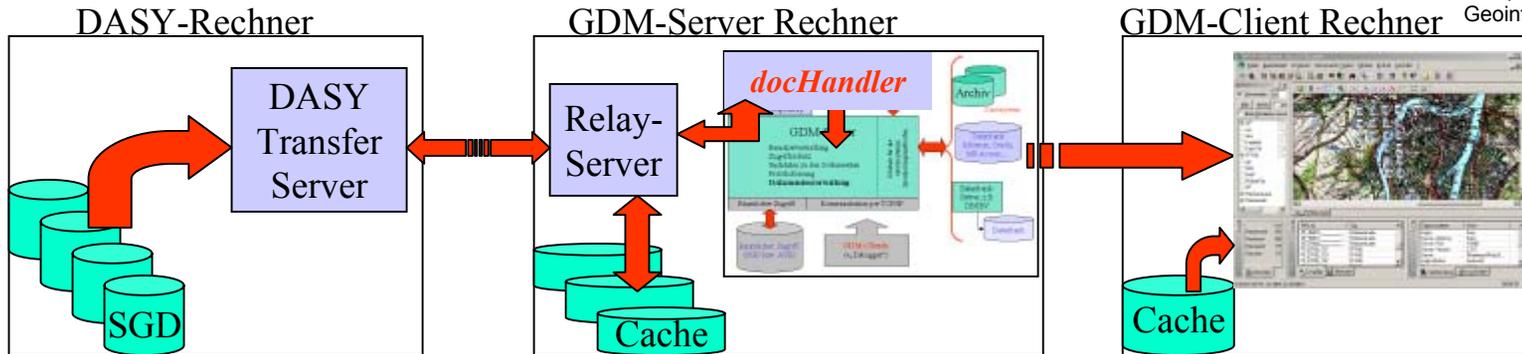
GDM – Serverbefehl hinzufügen

- Zeit für *Roundtrip* (Zeit zwischen GDM-Client-Request und Reply des Servers) beachten.
- Befehl eventuell *aufsplitten* auf kleinere Einzelbefehle.
- Ein Befehl kann aus mehreren *Subbefehlen* bestehen, der Befehlsname wird Bestandteil des Protokolls, die Subbefehle werden in der Aktionsroutine implementiert und abgeprüft.
- Berücksichtigung des *Zustandes* der Clientverbindung (accept, update, connect, ...).
- Festlegung welche *Rechte* dazu nötig sind.
- Einbringen des Befehls in **actionDefineCommands**: Befehlsliste erweitern um Name, Zustände vor und nach der Ausführung des Befehls und zugeordnete Aktionsroutine, die den Befehl implementiert.
 Konvention: Protokollbefehl 4-Großbuchstaben, falls Reply-Name nötig, wird dieser aus Reversierung des Befehlsnamens abgeleitet, z.B. ANNO => ONNA
- DB-Relationen: Falls der Befehl neue *DB-Relationen* benötigt, diese definieren und SQL-Skripte in **<server>/sql/base** erstellen.
- **actionNeuerBefehl**: Erstellen der *Aktionsroutine* in einer eigenen Tcl-Datei, die Datei ist im Unterverzeichnis **<server>/actlib** zu speichern. Der Tcl-Index ist zu aktualisieren
- **geoClient.tcl**: Nachziehen des Befehls für den Monitor zum Debuggen
- **geocmds.doc**: Befehl *dokumentieren* in der Liste der Serverbefehle.

GDM – Zugriff auf externe Dokumente (1)

- Alle Karten, die in **DASY** (Digitales Auskunftssystem LvermGeo Rheinland-Pfalz) verwaltet werden, sollen auch von **WEGA-GDM** verwendet werden.
- *Problem:*
 - Die Karten liegen als **QT**- oder **CBIM**-Raster in SGD-Dateien vor. SGD-Daten kann GDM jedoch nicht direkt anzeigen.
 - Es müssen die Dateien nach TIFF konvertiert werden (rechenzeitintensiv).
- Aufgrund der großen Datenmenge soll keine doppelte Datenhaltung der Karten für **WEGA-GDM** eingesetzt werden:
 - Die Dateien müssen daher beim Zugriff nach TIFF gewandelt und zum GDM-Client transferiert werden.
 - Um trotzdem akzeptable Antwortzeiten zu erhalten, wird am GDM-Server ein entsprechend großes Cache verwendet.
- *Problem:*
 - Es laufen in jeder Nacht Applikationsprogramme auf den Datenbestand, die größtenteils nichts ändern, aber den Zeitstempel der Dateien verändern, was zu unnötigen Transfers führen kann.

GDM – Zugriff auf externe Dokumente (2)



- GDM-Client fordert Dokument vom GDM-Server
- GDM-Server fragt per **docHandler** den **Relay-Server** ob Dokument vorhanden und eventuell auf aktuellem Stand
- Relay-Server fragt beim **DASY-GDM Transfer-Server** die Dokumentinformationen ab:
 - Prüfung, ob Dokument im Cache noch aktuell.
 - Falls vorhanden und aktuell, dann Freigabe an GDM-Server melden.
 - Ansonsten Transfer der SGD-Datei zum Server und Umsetzung nach TIFF oder Umsetzung der SGD-Datei nach TIFF auf dem DASY-Rechner und Transfer der fertigen TIFF-Datei (je nach Einstellung in Steuerdatei).
 - Transferierte TIFF-Datei ins Cache ablegen und Freigabe an GDM-Server melden

GDM – Neuen Server aufsetzen (1)

Es wird exemplarisch anhand eines Beispielservers aufgezeigt, welche Schritte notwendig sind, um einen neuen GDM-Server aufzusetzen:

1. Am besten man kopiert einen bereits vorhandenen Server

- neuer Name des Servers: **karlsruhe**
- Erstellen der Verzeichnisse
 - <server>/srvfiles/**karlsruhe**
 - <server>/sql/**karlsruhe**
- Kopieren und Anpassen der Dateien von existierenden Verzeichnissen

2. Zuständigkeitsgebiet festlegen: Gebiet Karlsruhe

- Eckkoordinaten links unten und rechts oben festlegen:
- diese Daten werden in der srv-Datei **karlsruhesrv.dat** in der Komponente **mcWorldBox** eingetragen.

3. SGD-Geobasis erstellen

- Mit Hilfe des EASI-Skripts **newsgd.tcl** kann eine „leere“ SGD-Datei erstellt werden:

```
easish
source newsgd.tcl
newSGD karlsruhe.sgd "3439400 5419500" "3475000 5442500"
file copy karlsruhe.sgd karlsruhe.org ;# Org-Datei für Recovery
```

- Die neue SGD-Datei enthält dann die Paßpunkte und die Attributtabelle für GDM

GDM – Neuen Server aufsetzen (2)

4. Übersichtskarte(n) erstellen oder beschaffen

- Es sind verschiedene Rasterformate möglich, z.B. [karlsruhe.jpg](#).
- Die Karte muß georeferenziert sein, z.B. GeoTIFF, oder diese Information wird in der srv-Datei in den Komponenten „[overviewFile](#)“ und „[overviewFile](#), [worldBox](#)“ hinterlegt.
- Es können auch beide Methoden verwendet werden (World-File: [karlsruhe.jpgw](#)) und Eintrag in srv-Datei.

5. srv-Datei „ausbauen“, indem man die Komponenten editiert

- Vergabe einer eindeutigen ServerID (eindeutig innerhalb der Datenbank): [serverID=98](#)
- Lizenzdaten eintragen in den Komponenten [licenseOwner](#), [licenseCode](#) und [licenseFeat](#) (die Lizenzdaten können mit dem Lizenzierungstool ermittelt werden).

6. Datenbank aufsetzen

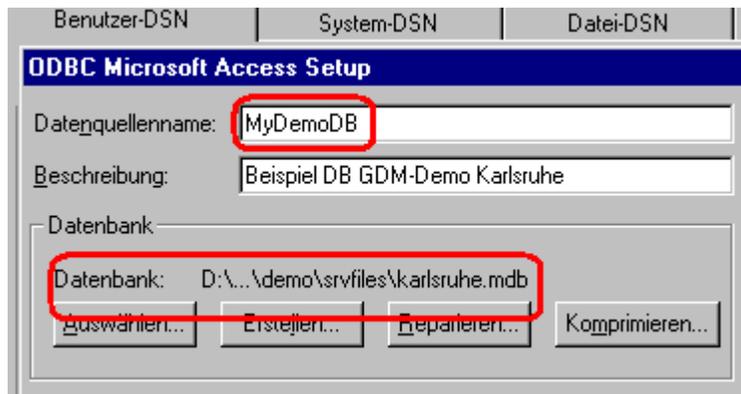
- Festlegen, welche Datenbank (ODBC, Oracle, Informix, ...), dadurch wird festgelegt, welches DB-Package der Server laden muß
- je nach „leere“ Datenbank/DB-Space/etc. anlegen
- je nach DB, Festlegen des DB-Zugriffs, z.B. ODBC-Datenquelle oder DB-Servername
- der [GDM-Server](#) sollte die entsprechenden Zugriffsrechte erhalten (u.a. Tabellen anlegen und löschen, Datensätze anlegen, modifizieren und löschen, ...)
- das Tcl-Skript [dbinitialize.tcl](#) anpassen, damit hier sofort auf die Datenbank zugegriffen wird: Variable [dbName](#) anpassen.

GDM – Neuen Server aufsetzen (3)

noch 6. Datenbank aufsetzen

- für unser Beispiel: ODBC-Verbindung MyDemoDB mit Access-Datenbank

karlsruhe.mdb



Dialog erreichbar per:
Start/Einstellungen/Systemsteuerung
→ Auswählen „Datenquellen (ODBC)“

Neue Datenquelle mit MS-Access-Treiber hinzufügen (wenn Server als Dienst laufen soll, dann System-DSN, sonst reicht auch Benutzer-DSN)

→ System-DSN „**MyDemoDB**“ hinzufügen

In ODBC-Setup, dann Datenbank auswählen bzw. erstellen

- nun Testen der Verbindung zu der neu erstellten Datenbank mit Hilfe des Tcl-Skriptes

dbinitialize.tcl im Verzeichnis <server>/sql

```
tclsh
```

```
source dbinitialize.tcl
```

```
# es sollten keine Fehler erscheinen
```

GDM – Neuen Server aufsetzen (4)

noch 6. Datenbank aufsetzen

- wenn alles OK, dann können wir die GDM-spezifischen Tabellen erstellen und füllen

```
tclsh
source dbinitialize.tcl
cd <server>/sql/base
createDB 1                ;# 1 Tabellen werden erzeugt, 0 Tabellen werden nur befüllt
sqlDisconnect
exit
```

- es erscheinen nun Informationen, welche Tabellen erstellt und mit welchen Daten geladen wurden. Hier ist zu prüfen, ob alle Tabellen und Daten korrekt geladen wurden.
- alle Ladeaktionen sollten vor Ende des Programms per `sqlDisconnect` „bestätigt“ werden.

7. Server-spezifische SQL-Daten

- Es muß in dem Unterverzeichnis `<server>/sql/karlsruhe` das Tcl-Skript „`dosysdatatvalues.tcl`“ vorhanden sein, bzw. editiert werden.
- in dieser Datei die `ServerID` und Serverkennung eintragen und wie bei 6. laden:

```
set values {
  { 98, 1, 1, 1, 'GDM-Server Karlsruhe', 'karlsruhe', 0, 0, 0, 0, 11, '1' }
}
....
tclsh
source dbinitialize.tcl
cd <server>/sql/karlsruhe>
source dosysdatatvalues.tcl                ;# ← ServerID laden
```

GDM – Neuen Server aufsetzen (5)

8. In der srv-Datei alle Pfadangaben und DB-Verbindung eintragen
 - Überprüfen der Komponenten `serverID` und `overviewFile`
 - in `dbConnect` wird die Datenbankverbindung hinterlegt,
 - `dbConnect = {ODBC MyDemoDB}`
 - danach sind die Pfade zu den Dateien festzulegen
 - `applPath`: definiert, wo sich die SGD-Datei, Protokolldatei, ... befinden
 - `filePath`: definiert, wo die eingerichteten Dateien zu speichern sind
(nur wenn Speicherung im Dateisystem)
 - Dateinamen für SGD-Datei, Protokolldatei und SGD-Masterdatei (=org-Datei) definieren
 - `logfile`, `sgdFilename` und `sgdMasterFile`
 - eventuell notwendige spezifische Tcl-Skripte und Prozeduren definieren
 - `srvSources`: die angegebenen Quellen werden sofort geladen (in Phase 1)
 - `file2ppGenerator`: Prozedurname zum Umsetzen von Dateinamen auf Paßpunkte
 - `dokidGenerator`: Prozedurname zum Generieren einer eindeutigen Dokumentenkennung (DokID) anhand von Sachdaten und Dokumententyp

GDM – Neuen Server aufsetzen (6)

9. Unter Unix Anpassen des *Enable*-Skripts „wega.enb_ksh“

- An sich ist alles in der srv-Datei definiert, jedoch benötigen einige DB-Systeme noch Umgebungsvariablen, diese kann man entweder
 - in einem eigenen Skript definieren, dann muß dieses Shellskript aber vor dem Enable-Skript ablaufen, z.B. Definitionen im Login-Skript „.profile“
 - oder im Unterverzeichnis „setup“ des Servers sind in dem Shellskript „wega.enb_ksh“ die Umgebungsvariablen zu definieren (Vorsicht, kann bei Updates überschrieben werden)
- festlegen, welche srv-Datei verwendet werden soll: **WEGA_GDMPARAMFILE**

10. Testen, ob der Server „hochfährt“

- Terminal mit entsprechender Umgebung aktivieren, z.B. **WEGAShell** oder **RoSyShell** unter Win32, oder neues Login unter Unix.
- zum Serververzeichnis wechseln und den Server starten

```
cd <server>
server start srvfiles/karlsruhe/karlsruhesrv.dat ← Win32
StartServer -console                               ← Unix
```

- es sollte danach der Server wieder heruntergefahren werden per

```
server stop srvfiles/karlsruhe/karlsruhesrv.dat ← Win32
^Z                                               ← Unix
ShutServer                                       ← Unix
```

GDM – Neuen Server aufsetzen (7)

11. Analysieren der Ausgabe am Terminal und der Protokolldatei

- Hinweise, ob der Zugriff auf die Datenbank korrekt ist, und ob der richtige Server aktiviert wurde.
- Prüfung, ob Lizenzierung in Ordnung ist.
- Prüfung, welche Features aktiviert sind:

```
02/09/18 13:18:51 Logging of process 217 started: WEGA-GDM GeoServer
02/09/18 13:18:51: (setup) Starting geoserver in directory 'D:/Wega-GDM/server,
02/09/18 13:18:51: (setup) Try to connect to database...
02/09/18 13:18:51: (setup) ODBC connect to MyDemoDB
02/09/18 13:18:51: (setup) Try to connect to sgdfilere...
02/09/18 13:18:52: (criti) Server ID 98 Name 'GDM-Server Karlsruhe' with (1, 1)
02/09/18 13:18:52: (setup) Checking License...
02/09/18 13:18:52: (setup) License yields 20 max. Clients
02/09/18 13:18:52: (setup) Activated Features GeoTIFF,CPTdialog,Attachments
02/09/18 13:18:52: (setup) Setup listener socket...
```

12. Mit GDM-Client Verbindung testen

- Wenn alles soweit o.k. ist oder korrigiert worden ist, dann den Server wieder starten
- GDM-Client aktivieren und verbinden zum Server
 - Prüfung des Verbindungsaufbaus: Rechnername und Portnummer
 - Prüfung, ob Overview erscheint und die angezeigten Koordinaten plausibel

GDM – Neuen Server aufsetzen (8)

13. Aufsetzen der Clientskripte und evtl. der OCX-basierten Formulare

- am besten Skripte und OCX-Dateien von existierender Umgebung holen

→ z.B. `<server>/clntscripte/democd` → `<server>/clntscripte/karlsruhe`

→ alle unnötigen Dateien entfernen und VBS-Dateien anpassen, insbesondere die Dateien

- + `init.vbs`
- + `login.vbs`
- + `add_hit.vbs`

- zuerst sollte man mit den dynamischen Formularen beginnen

→ `GDM_DYNAMISCH.ocx`

14. Pfad zu den Clientskripten in srv-Datei prüfen/anpassen

- Eintrag „`lappend geoSetup(clientScriptsPath) [file .../karlsruhe]`“

15. Server neu starten und dann mit Client Verbindung aufnehmen

- es sollten die Clientskripte beim Login zum Client transferiert werden
- prüfen, ob OCX richtig registriert, indem man die Suchmaske aktiviert

GDM – Neuen Server aufsetzen (9)

16. Server als Dienst unter Win32 einrichten

- Datei GDMServerDemo.exe kopieren nach **GDMServerKarlsruhe.exe**
- Datei GDMServerDemo.ini kopieren nach **GDMServerKarlsruhe.ini**
- Parameter in den Sektionen der Ini-Datei anpassen (Name Dienst und Pfade):

[System]

ServiceName=**GDM-Server-Karlsruhe**

....

[GDMStart]

Params=start srvFiles**karlsruhesrv.dat**

...

[GDMStop]

Params=start srvFiles**karlsruhesrv.dat**

- danach Service (Dienst) installieren und starten

WEGAShell oder RoSyShell aktivieren

```
cd <Pfad, wo GDMServerKarlsruhe.exe sich befindet>
```

```
GDMServerKarlsruhe.exe -i ; Dienst installieren
```

```
GDMServerKarlsruhe.exe -s ; Dienst starten
```

GDM – Neuen Server aufsetzen (10)

17. Überprüfen, ob Dienst aktiv

- Unter Systemsteuerung den Dialog „Dienste“ aktivieren und prüfen, ob der Dienst in der Liste vorhanden ist.



18. Weitere Dokumenttypen einbringen

- Mit Hilfe des Tcl-Skriptes `dotypdefvalues.tcl` die SQL-Tabelle `dotypdef` entsprechend erweitern.
- zugeordnete Sachdatentabellen definieren und als Tcl-Skripte `<tabellenname>.tcl` im Unterverzeichnis `<server>/sql/karlsruhe` speichern, zB. `hotelsd.tcl`
- wenn Sachdaten vorhanden, dann Dictionary-Tabellen `tblWegaGDM` und `tblWegaColumns` erweitern und in den zugeordneten TCL-Skripten speichern
- danach sollte der Server neu gestartet werden!

19. Server und Client weiter „ausbauen“

- Applikationen, Formulare, EASI/Tcl-Prozeduren