



# Datenaustausch in der Raumordnung mit XPlanung

Robert Krätschmer  
Christian Seip  
Peter Korduan  
**GDI-Service Rostock**



- Einleitung
- XPlanung
- Erweiterung des Modells
- Transformation in eine Datenbankstruktur
- Ausblick



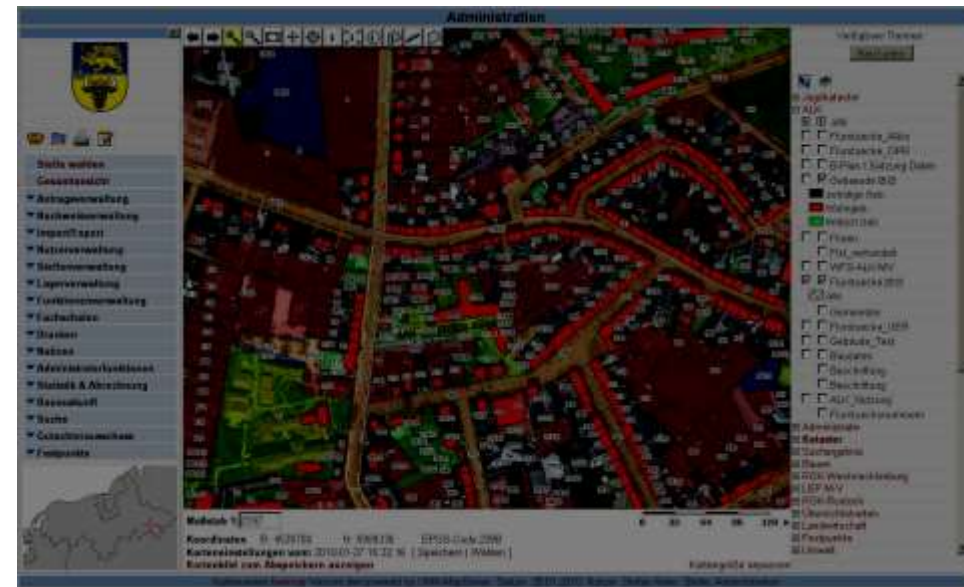
# Einleitung

# GDI Service Rostock

Beratung – Entwicklung – Schulung – Dienste – Hosting



## Fahrzeug Online Tracking



## WebGIS kvwmap

# Einleitung

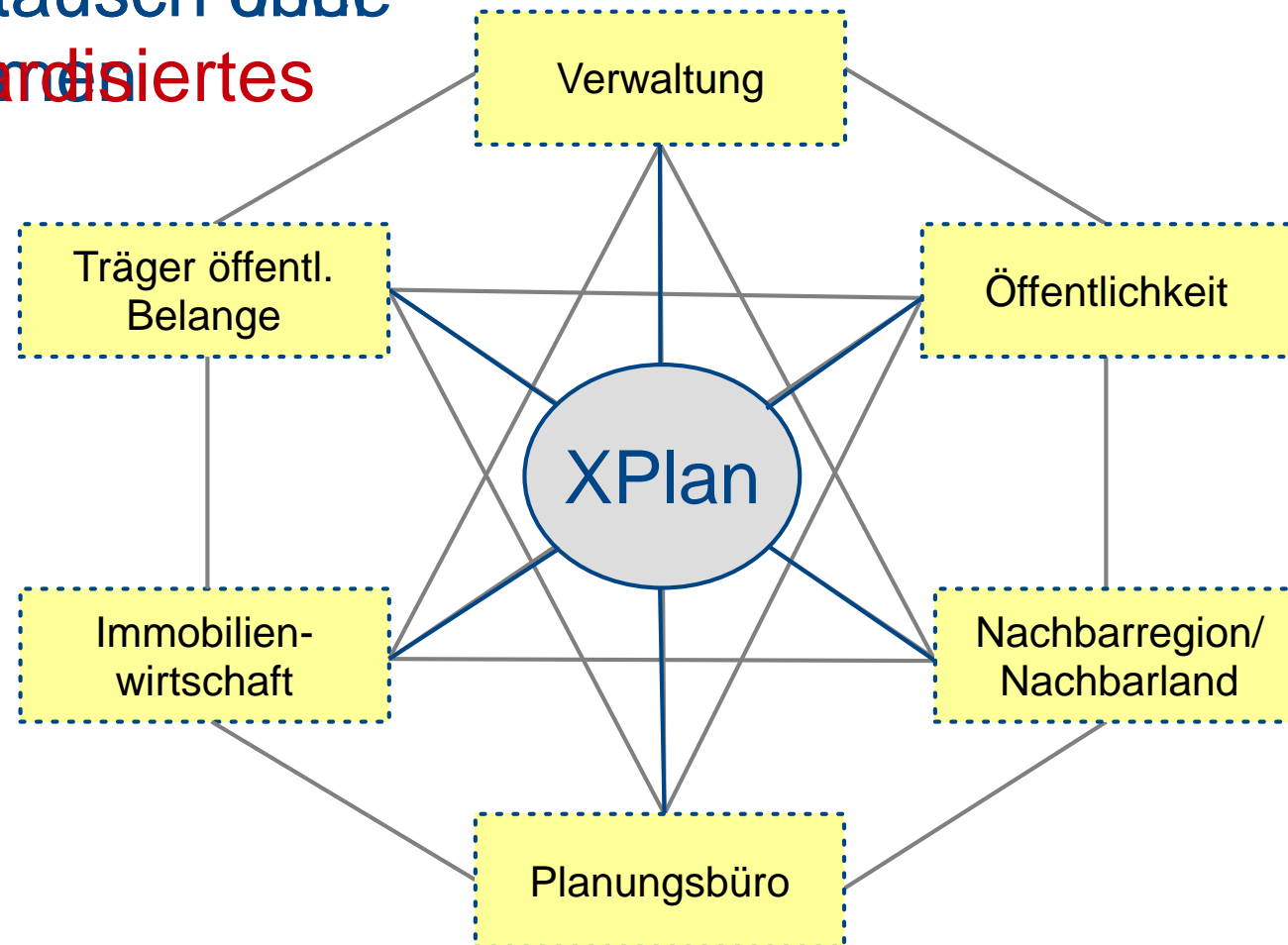


- Kein einheitliches Format für Geodaten  
von **Raumordnungsplänen**

# Einleitung



Datenaustausch über  
ein **standardisiertes**  
Standard



# Einleitung



• Modellvorhaben der Raumordnung „Entwicklung und Implementierung eines Standards für den Datenaustausch in der Raumordnungsplanung“

Weiterentwicklung von XPlan

Berücksichtigung von INSPIRE

Konverterentwicklung

Dokumentation & Ontologie





# XPlanung





# XPlanung



- Austausch durch
- XPlan-GML
- (XML-Grammatik)

```
<xml version="1.0" encoding="utf-8"?>
<XplanAuszug xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" ...>
  <gml:featureMember>
    <RP_Plan gml:id="GML_2852f398-29d6-496c-979d-37fd38cdfab4">
      <xplan:name>Musterregionalplan-Name</xplan:name>
      <xplan:bundesland>1000</xplan:bundesland>
    </RP_Plan>
  </gml:featureMember>
  <gml:featureMember>
    <xplan:RP_IndustrieGewerbe gml:id="GML_08190d1f-90e2-4d98-9889-059d0ca805e0">
      <xplan:typ>1000</xplan:typ>
      <xplan:position>
        <gml:Polygon srsName="EPSG:31466">
          <gml:exterior>
            <gml:LinearRing>
              <gml:pos>2518000.000 5636730.000</gml:pos>
              <gml:pos>2517961.000 5636726.000</gml:pos>
              <gml:pos>2517932.000 5636722.000</gml:pos>
            </gml:LinearRing>
          </gml:exterior>
        </gml:Polygon>
      </xplan:position>
    </xplan:RP_IndustrieGewerbe>
  </gml:featureMember>
  ...
</XplanAuszug>
```



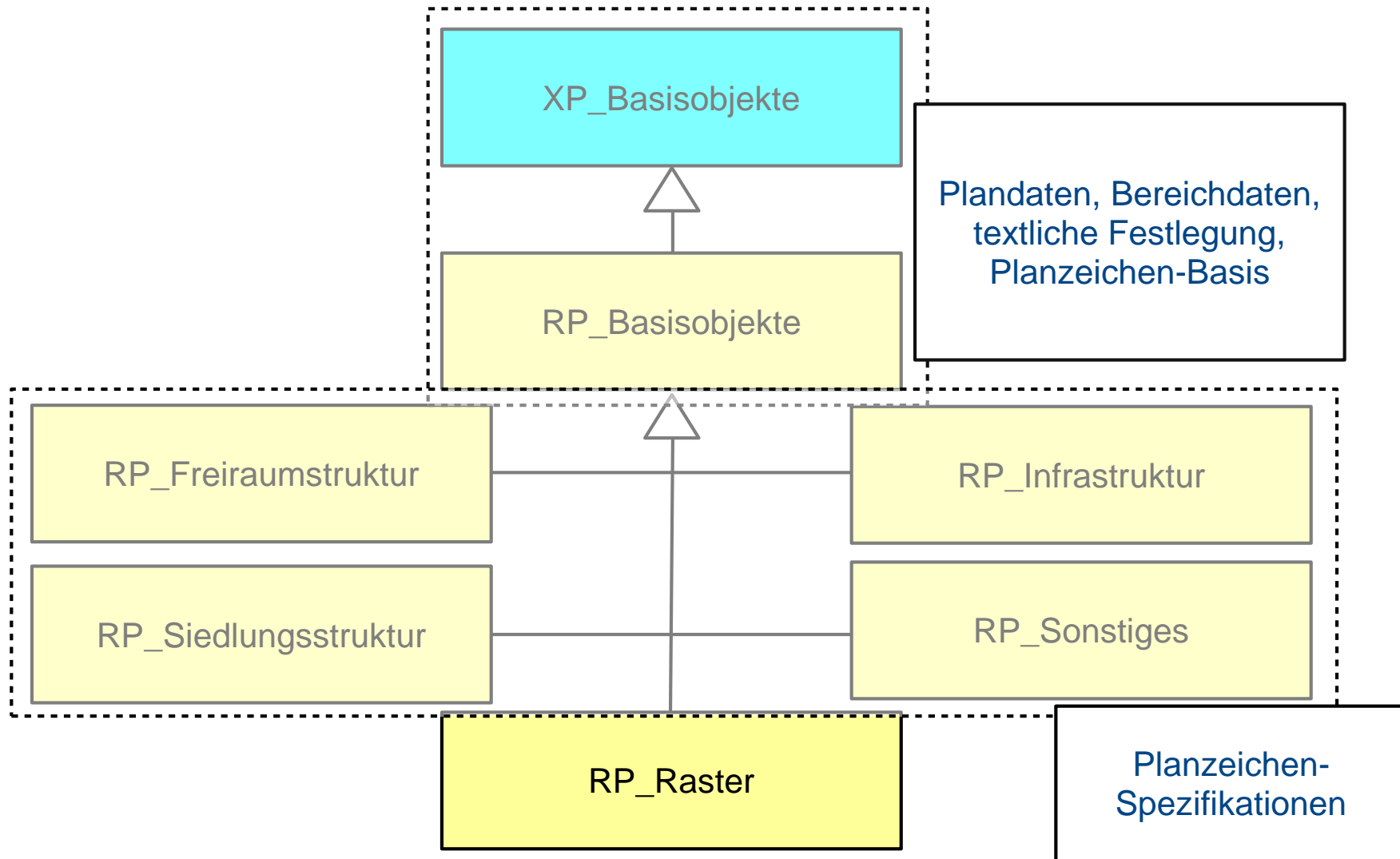
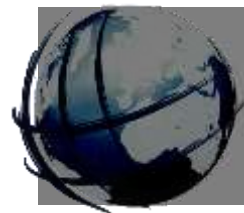
• XPlan ist:

Objektorientiert

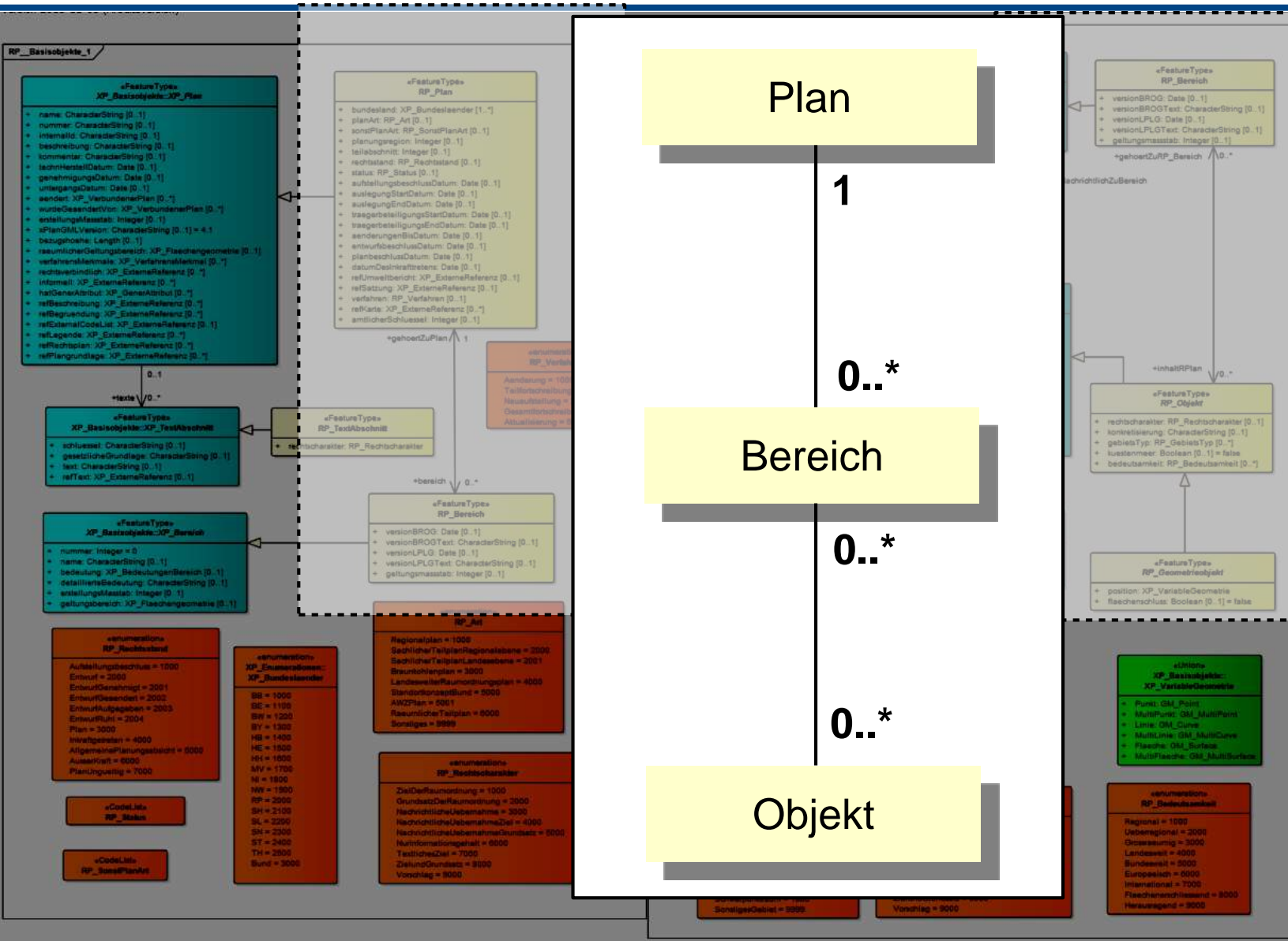
Verlustfrei

Planübergreifend

# XPlanung



# XPlanung





# Erweiterung des Modells



# Erweiterung des Modells



- Erweiterung auf Basis von:  
NRW-, NSM-, RLP-Länderschema  
ROPLAMO  
INSPIRE  
Ländergespräche  
Gespräche mit der AG  
E-Government der MKRO  
weiterer Quellen



# Erweiterung des Modells



3

«FeatureType»  
**RP\_ZentralerOrt**

- + typ: RP\_ZentralerOrtTypen [1..\*]
- + sonstigerTyp: RP\_ZentralerOrtSonstigeTypen [0..\*]

«enumeration»  
**RP\_ZentralerOrtTypen**

Oberzentrum = 1000  
GemeinsamesOberzentrum = 1001  
Oberbereich = 1500  
Mittelzentrum = 2000  
Mittelbereich = 2500  
Grundzentrum = 3000  
Untzentrum = 3001  
Nahbereich = 3500  
Kleinzentrum = 4000  
LaendlicherZentralort = 5000  
Stadtrandkern1Ordnung = 6000  
Stadtrandkern2Ordnung = 6001  
VersorgungskernSiedlungskern = 7000  
ZentralesSiedlungsgebiet = 8000  
Metropole = 9000  
SonstigerZentralerOrt = 9999

«enumeration»  
**RP\_ZentralerOrtSonstigeTypen**

Doppelzentrum = 1000  
Funktionsteilig = 1100  
MitOberzentralerTeilfunktion = 1101  
MitMittelzentralerTeilfunktion = 1102  
ImVerbund = 1200  
Kooperierend = 1300  
KooperierendFreiwillig = 1301  
KooperierendVerpflichtend = 1302  
ImVerdichtungsraum = 1400  
SiedlungsGrundnetz = 1500  
SiedlungsErgaenzungsnetz = 1501  
Entwicklungsschwerpunkt = 1600  
Ueberschneidungsbereich = 1700  
Ergaenzungsfunktion = 1800  
Nachbar = 1900  
MoeglichesZentrum = 2000  
SonstigeSonstigerZentralerOrt = 9999



# Transformation des Modells in eine Datenbankstruktur





# Ausgangslage

---

• XPlanGML wird modelliert

- mit UML (Unified Modeling Language)

- in Enterprise Architect (proprietär und kostenpflichtig)

• d. h. das Modell kann ausschließlich in Enterprise Architect bearbeitet werden



# Lösung

- Austauschformat XMI (XML Metadata Interchange)
- XMI kann auch in Datenbankschema überführt werden:
  - Nachverfolgbarkeit von Änderungen
  - Austausch von Dateien entfällt
  - Aufbereitung der Modellelemente (also beispielsweise der Klassen mit ihren Attributen und Assoziationen) im Browser



•nötig

◦bei neuen Umgebungen, Architekturen und Modellierungssprachen

◦für Modularität, einfachere Wartbarkeit und vereinfachtes Debugging der Modelle

•zwei Herangehensweisen:

•1. Modell-zu-Text Transformation (M2T, auch als Modell-zu-Code bezeichnet)

•2. Modell-zu-Modell-Transformation (M2M)



# Entwicklungsrahmen

.PHP

.XMI => SQL-Skript zum füllen der Datenbank

.Angaben:

- o (neues oder vorhandendes) Schema, eventuell Erzeugung der Tabellenstruktur

- o ob man ein spezifisches Paket (package) und dessen Unterpakete transformieren möchte oder alle

.Optionen:

- o truncate sorgt bei bestehendem Schema dafür, dass alle Tabellen zunächst geleert werden

- o „Argo Export mit ISO19136 Profil“



# Oberfläche

## Dateiauswahl

..

2016-01-18\_Modell\_EA-xmi12-uml14.xmi

2016-01-21\_Modell\_AAA-xmi12-uml14.xmi

## Schemaauswahl/-eingabe

db\_schema

## BasePackageauswahl/-eingabe

Bei einem EA-xport unbedingt "XPlanGML 4.1" wählen, bei einem ArgoUML Export leer lassen oder ein Package eintragen, falls man nur das eine laden möchte.

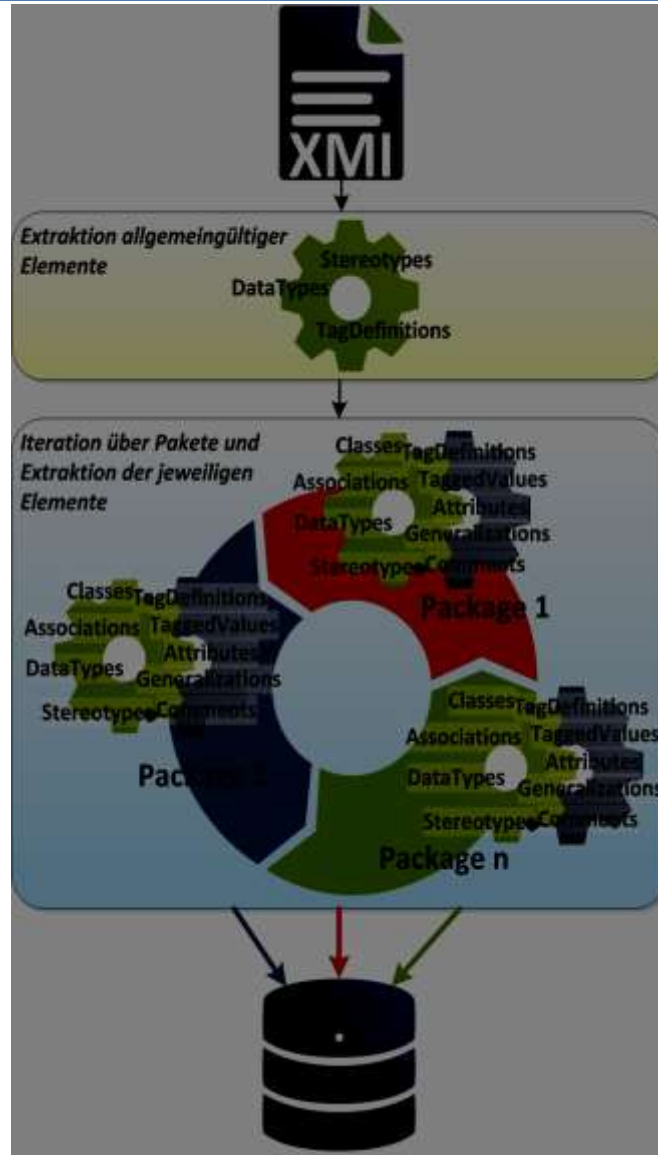
XPlanGML 4.1

- truncate
- Argo Export mit ISO19136 Profil

Fülle DB mit XMI Inhalten



# Transformationsprozess



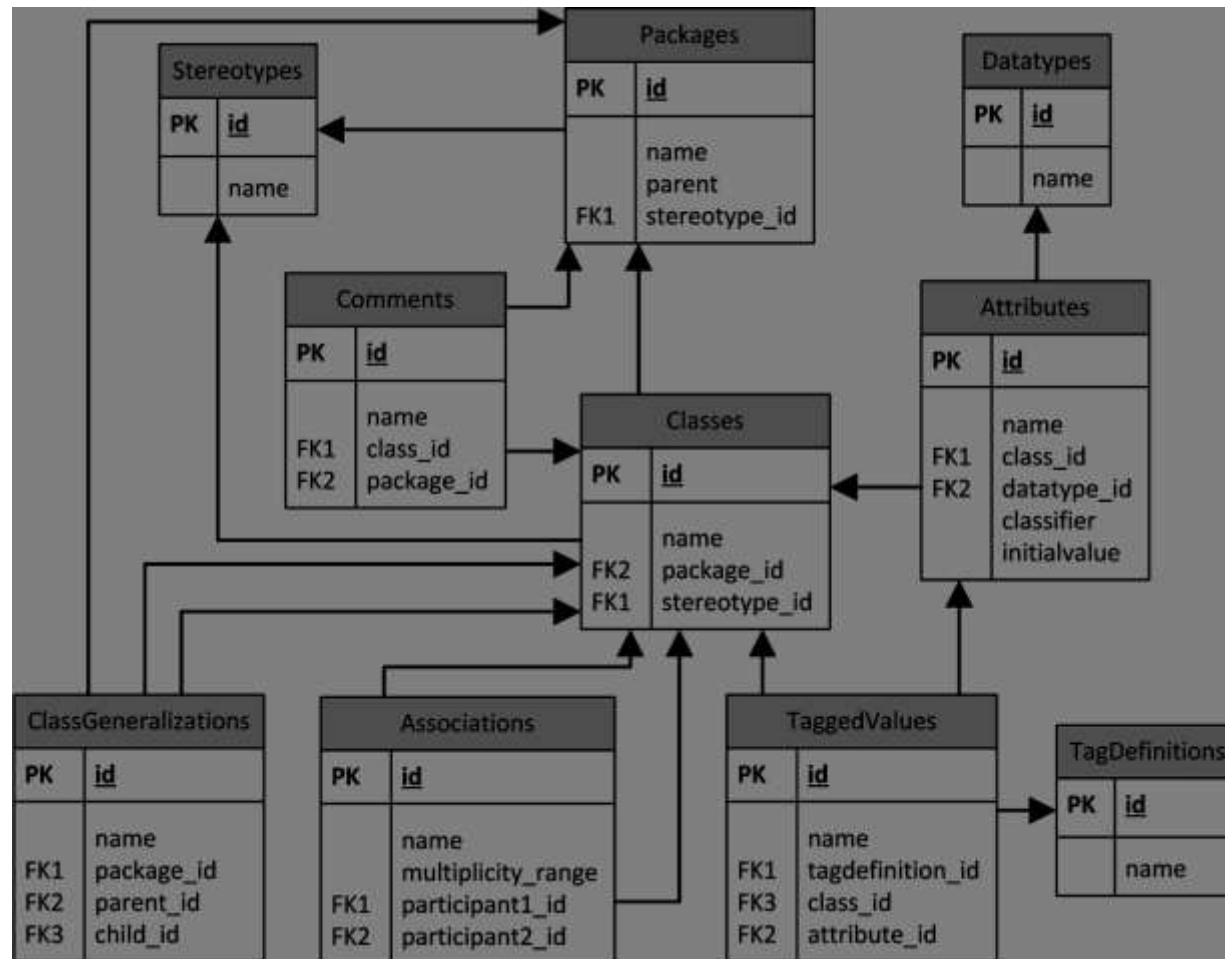


# Transformationsprozess

1. global definierte Datentypen, Stereotypen und TagDefinitionen
2. Iteration über Pakete des Modells
  - Paket selbst (Tabelle *Packages*)
  - Klassen des Pakets (Tabelle *Classes*)
    - Attribute (mit einfachem oder komplexen Datentyp, Multiplizitätsbereich, initialer Wert usw.)
    - Beziehungen zwischen den Klassen (XP\_Plan Generalisierung von RP\_Plan, Tabelle *Generalizations* oder RP\_Plan assoziiert mit RP\_Bereich, Tabelle *Associations*)
  - Kommentare Pakete und Klassen (Tabelle *Comments*)



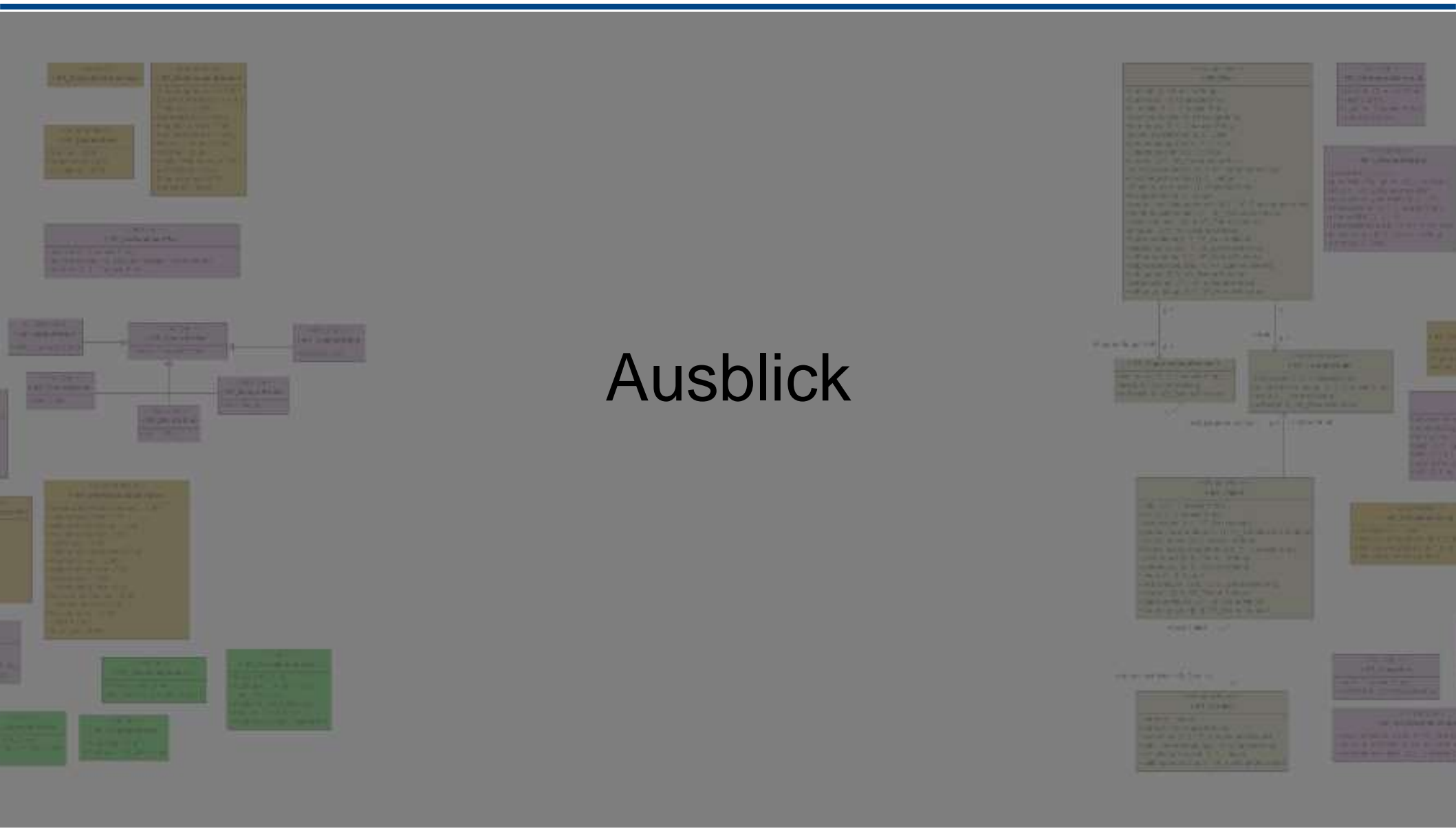
# Datenbankschema







# Ausblick





- Schnittstelle XPlanGML/INSPIRE
- Softwarelösung & Tests
- Ontologie & Dokumentation
- Visualisierung