

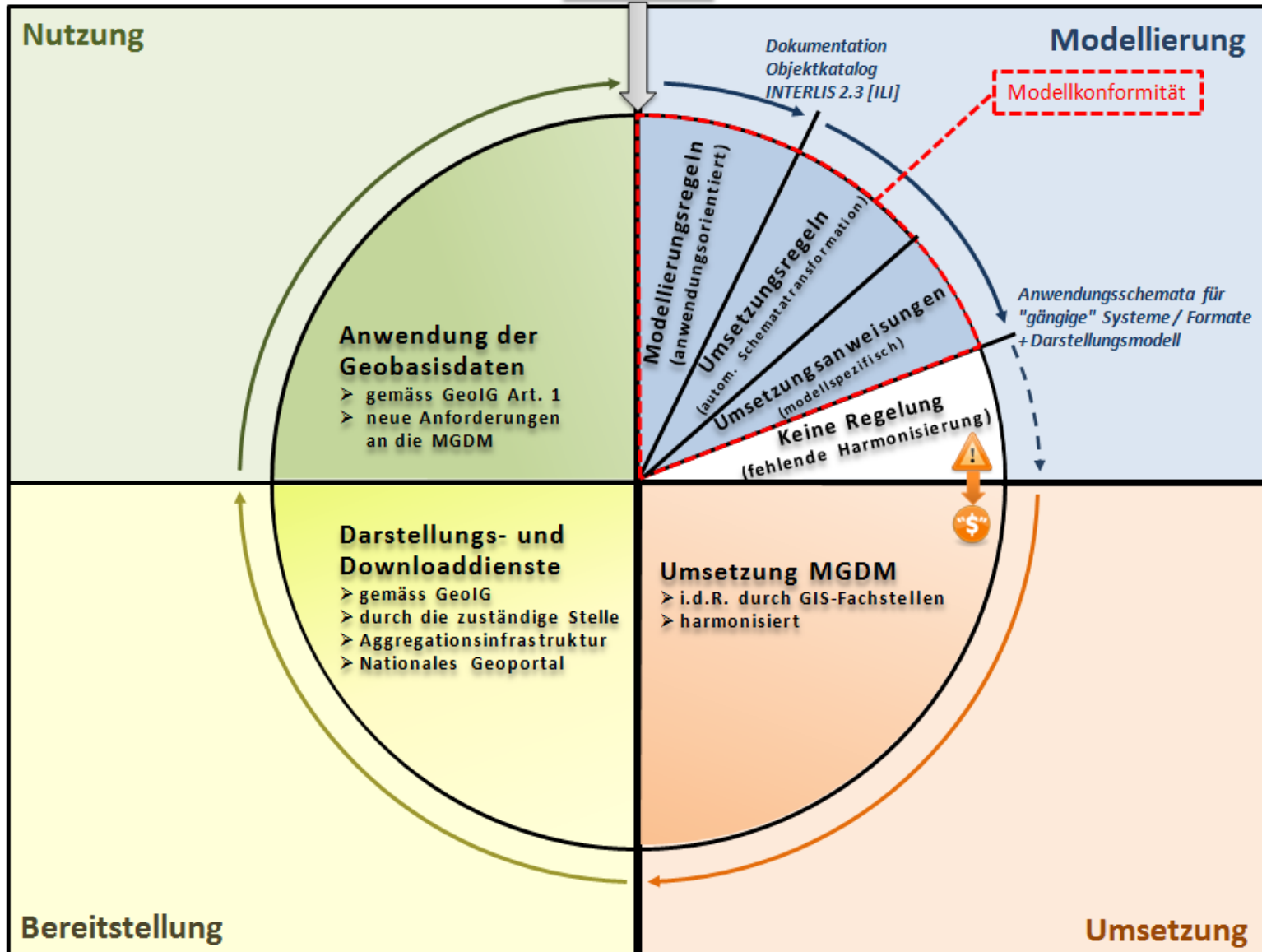
Abgabe und Übernahme von MGDM Daten durch gängige Systeme, modellbasiert

(Stand der Diskussion in der SOGI Fachgruppe 3)

Martin Stahl
Hans Rudolf Gnägi
Sigi Heggli
Christine Giger

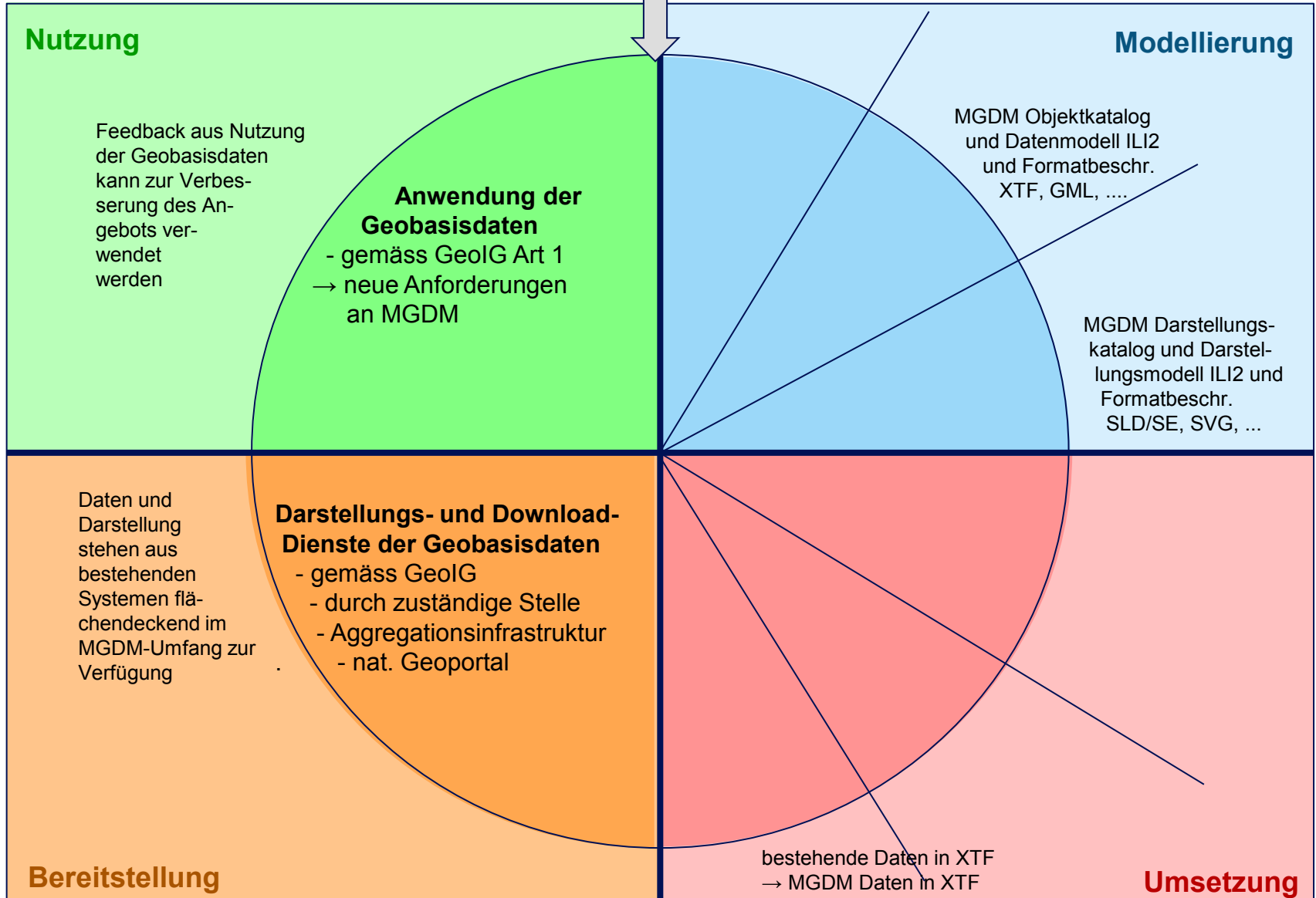
Erarbeitung und Anwendung der MGDM

Fachgesetz



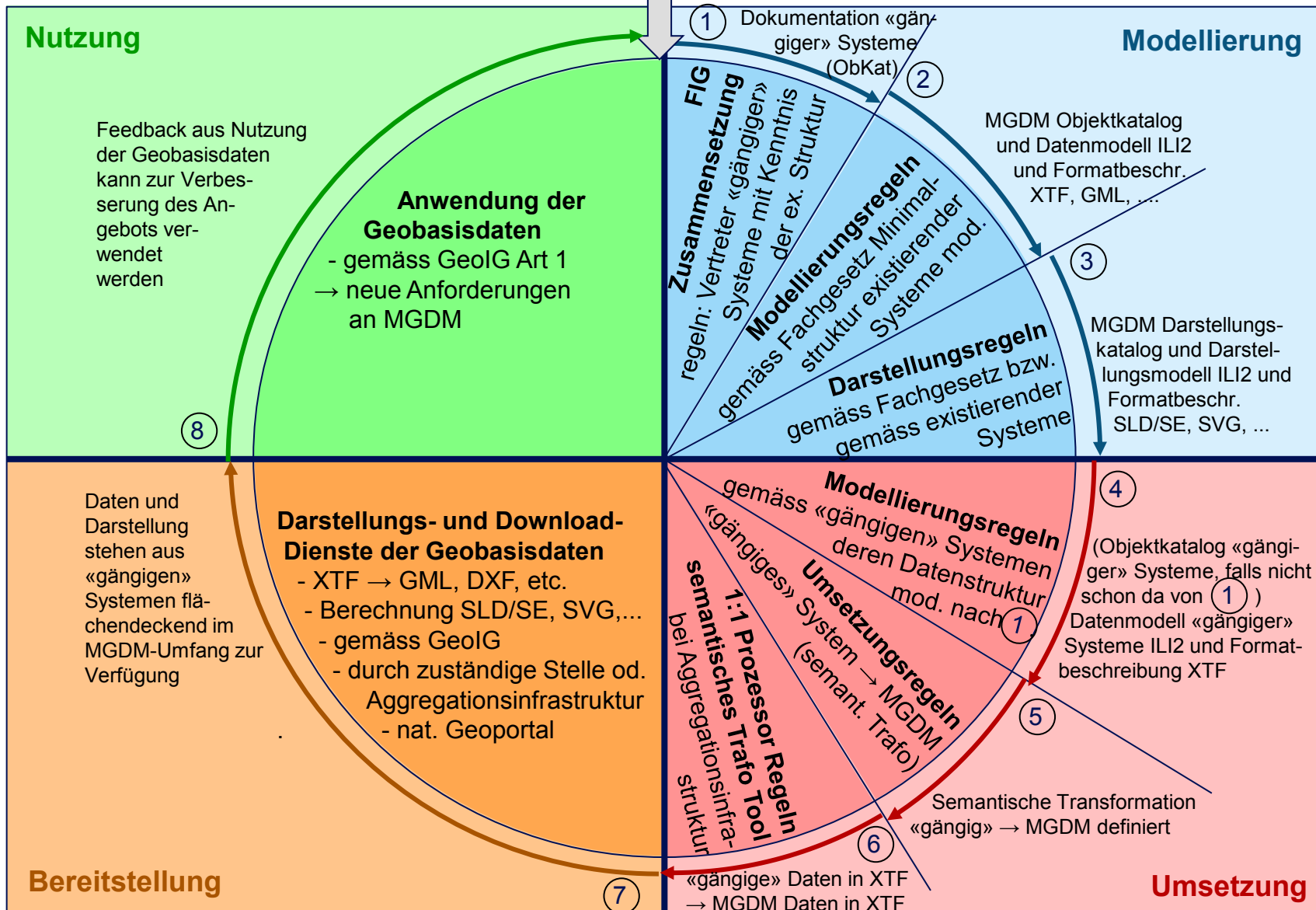
Erarbeitung und Anwendung der MGDM

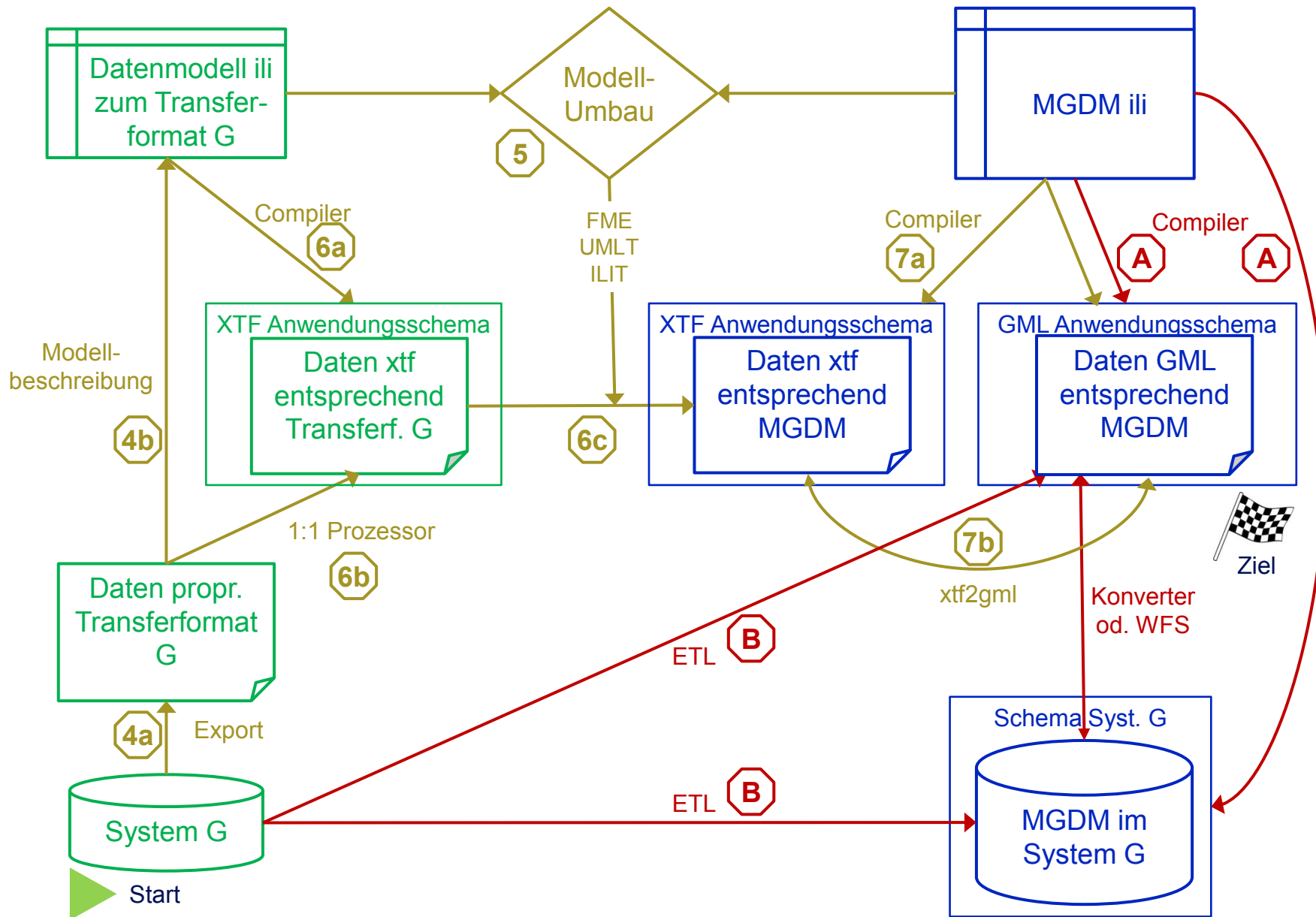
Fachgesetz



Erarbeitung und Anwendung der MGDM

Fachgesetz





Stand der Diskussion in SOGI Fachgruppe 3

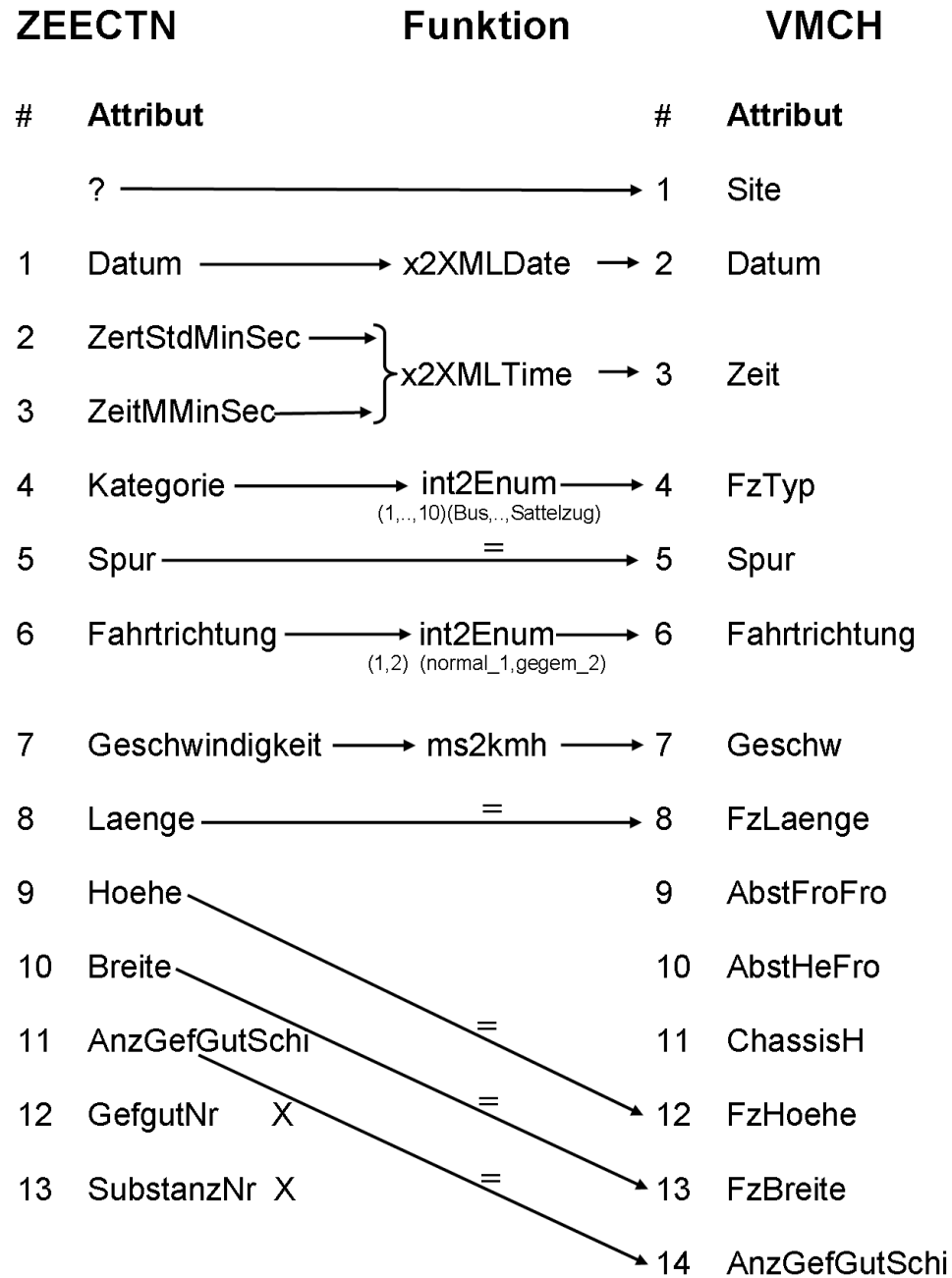
- Modellbasiertes Vorgehen optimal auch für Umsetzung der MGDM, d.h. für Abgabe und Übernahme von MGDM Daten
 - einfache funktionierende Grundlage in CH: INTERLIS und UML
 - system-abhängige Programmierung minimal (1:1 Prozessor)
 - Programmierkapazität einsetzen für Werkzeuge, die wieder verwendbar sind (UMLT, ILIT, UML-INTERLIS Editor, Compiler,...)
- FG 3 soll zeigen, dass es funktioniert → Pilotprojekt
 - Projektskizze formuliert
 - Finanzierung GeoWave von SOGI?
- Realisierungsallianzen

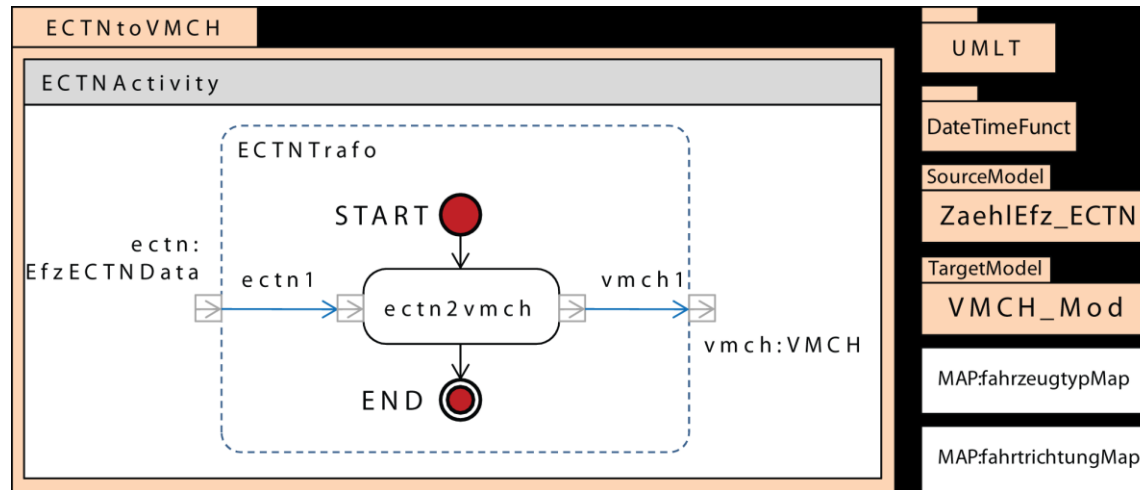
EfzECTNData

Datum[1] : Zeichenkette
 ZeitStdMinSek[1] : Zeit
 ZeitMiliSek[1] : Numerisch
 Kategorie[1] : Numerisch
 Spur[1] : Numerisch
 Fahrtrichtg[1] : Zeichenkette
 Geschwindkt[1] : Numerisch
 Laenge[1] : Numerisch
 Hoehe[1] : Numerisch
 Breite[1] : Numerisch
 AnzGefGutSchi[1] : Numerisch
 GefahrgutNr[0..1] : Numerisch
 SubstanzNr[0..1] : Numerisch

VMCH

Site[0..1] : Numerisch
 Datum[1] : XMLDate
 Zeit[1] : XMLTime
 FzTyp[1] : swiss10
 Spur[1] : Numerisch
 FahrRtg[1] : Aufzählung
 Geschw[1] : Numerisch
 FzLaenge[1] : Numerisch
 FzHoehe[0..1] : Numerisch
 FzBreite[0..1] : Numerisch
 AnzGefGutSchi[0..1] : Numerisch
 AbstFroFro[0..1] : Numerisch
 AbstHeFro[0..1] : Numerisch
 ChassisH[0..1] : Aufzählung







```
INTERLIS 2.3;
MAPPING MODEL ECTNtoVMCH (de) AT "www.gis.ethz.ch" VERSION "0.1" =

IMPORTS UNQUALIFIED UMLT;           !! Basisfunktionen
IMPORTS UNQUALIFIED DateTimeFunc;   !! Formatierungsfunktionen
IMPORTS ZaehlEfz_ECTN;               !! Quellmodell
IMPORTS VMCH_Mod;                   !! Zielmodell

TOPIC ECTNtoVMCH =

    ACTIVITY ECTNActivity =

        TRANSFORMATION ECTNTrafo =
            IN ectn: ZaehlEfz_ECTN.ZE_ECTN.EfzECTNData;
            OUT vmch: VMCH_Mod.VMCH_Top.VMCH;

        TRAFO_ACTION ectn2vmch =
            IN ectn1;
            OUT vmch1;

        MAP fahrzeugtypMap =
            1 -> "Bus";
            2 -> "Motorrad";
            3 -> "Personenwagen";
            4 -> "PWmitAnhaenger";
            5 -> "Lieferwagen";
            6 -> "LieferwagenMitAnhaenger";
            7 -> "LieferwagenMitAufleger";
            8 -> "Lastwagen";
            9 -> "Lastenzug";
            10 -> "Sattelzug";
        END fahrzeugtypMap;
```

```

MAP fahrtrichtungMap =
    "TRUE" -> "normal_1";
    "FALSE" -> "gegen_2";
END fahrtrichtungMap;

MAPPING
    vmch1->Datum := Transformer.ECTNtoXmlDate(ectn1->Datum);
    vmch1->Zeit := Transformer.ECTNtoXmlTime(ectn1->ZeitStdMinSek,
                                              ectn1->ZeitMilliSek);
    vmch1->FzTyp := ValueMapper.map(ectn1->Kategorie, fahrzeugtypMap);
    vmch1->Spur := ectn1 -> Spur;
    vmch1->FahrRtg := ValueMapper.map(ectn1->Fahrtrichtg, fahrtrichtungMap);
    vmch1->Geschw := ectn1->Geschwindkt*3.6;
    vmch1->FzLaenge := ectn1->Laenge;
    vmch1->FzHoehe := ectn1->Hoehe;
    vmch1->FzBreite := ectn1->Breite;
    vmch1->AnzGefGutSchi := ectn1->AnzGefGutSchi;

END ectn2vmch;

CONTROLFLOW
    START_FLOW --- ectn2vmch;
    ectn2vmch --- END_FLOW;

DATAFLOW
    ectn --- ectn2vmch.ectn1;
    ectn2vmch.vmch1 --- vmch;

    END ECTNTrafo;    !! Transformation
    END ECTNActivity; !! of Activity
END ECTNtoVMCH;      !! of Topic

END ECTNtoVMCH.

```