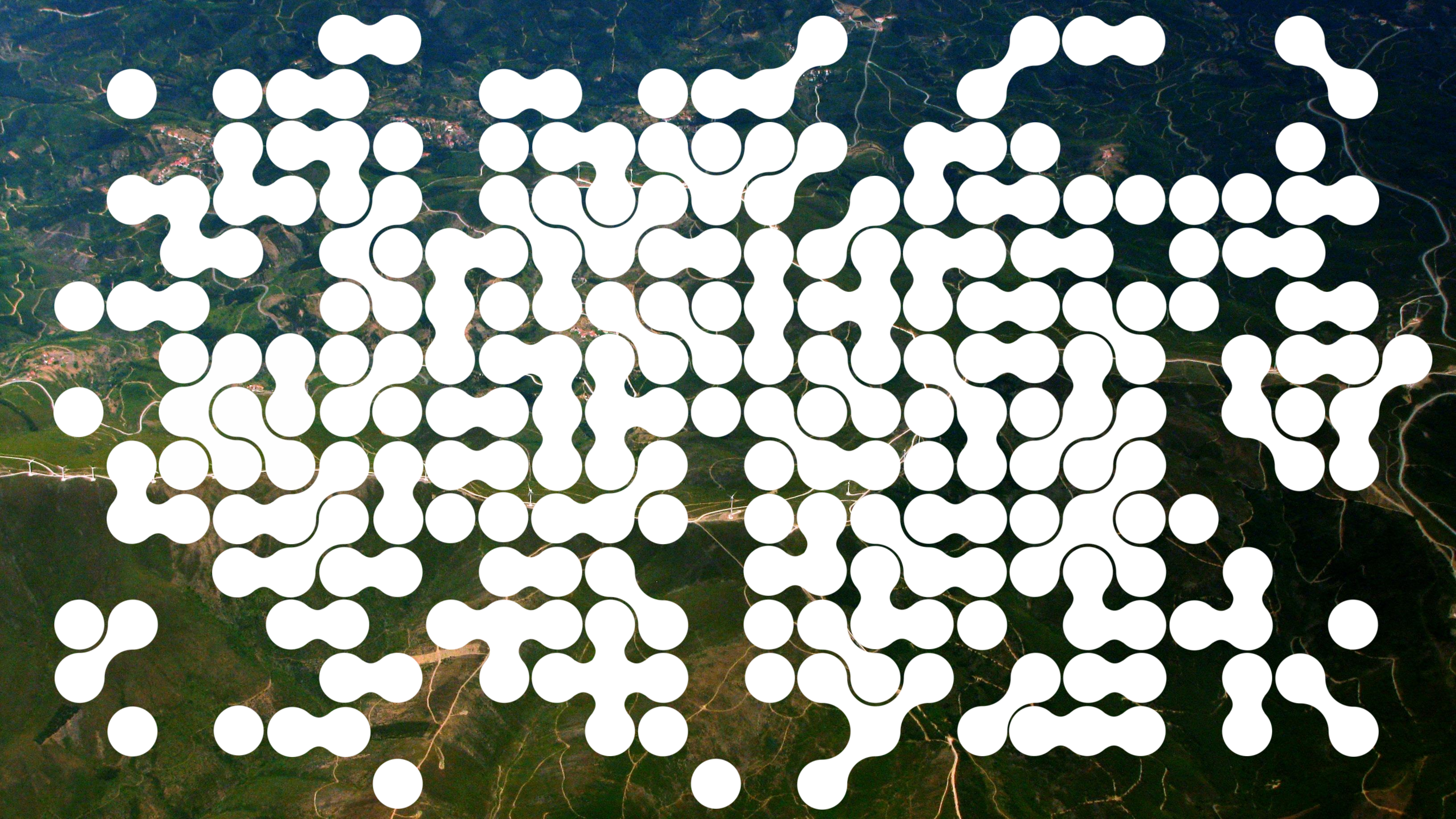


LandNetwork

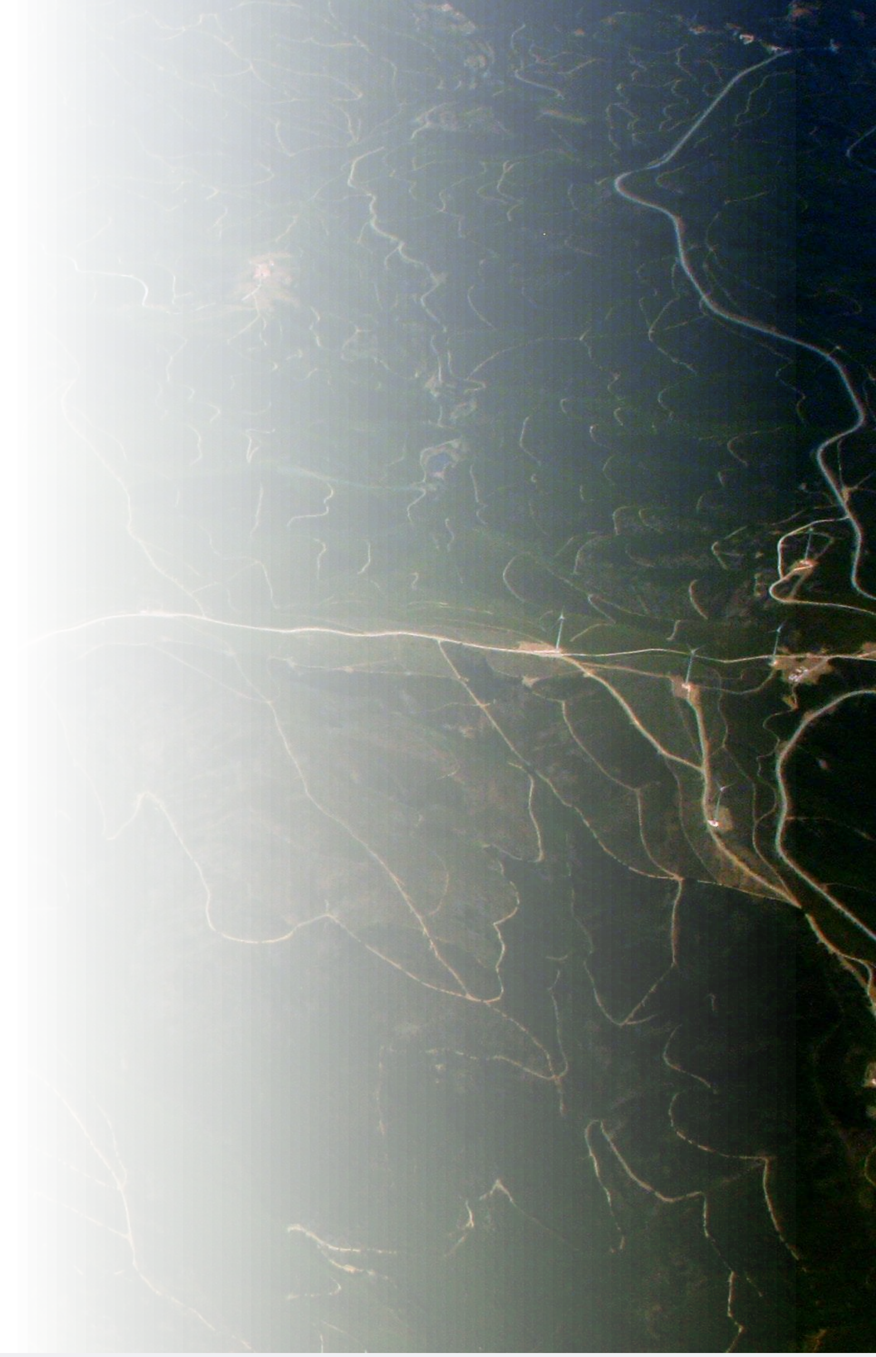
INTERLIS Training

Fabian Mejia
Land Administration Specialist
Fabian.mejia@landnetwork.ch

LandNetwork

Agenda

- External Catalogues
- Imports / Exports
  - Baskets
- Data Validation
  - Constraints
  - Functions
  - iliValidator TOML configuration and meta-attributes
- Pending questions



LandNetwork

Course Resources

Tools

- Text Editor: Notepad ++ (Recommended)
- QGis 3.16 +
- PostgreSQL 9.6 or upper + PostGIS 2.3 or upper
- Java VM (JRE 1.6 or upper)

Reference:

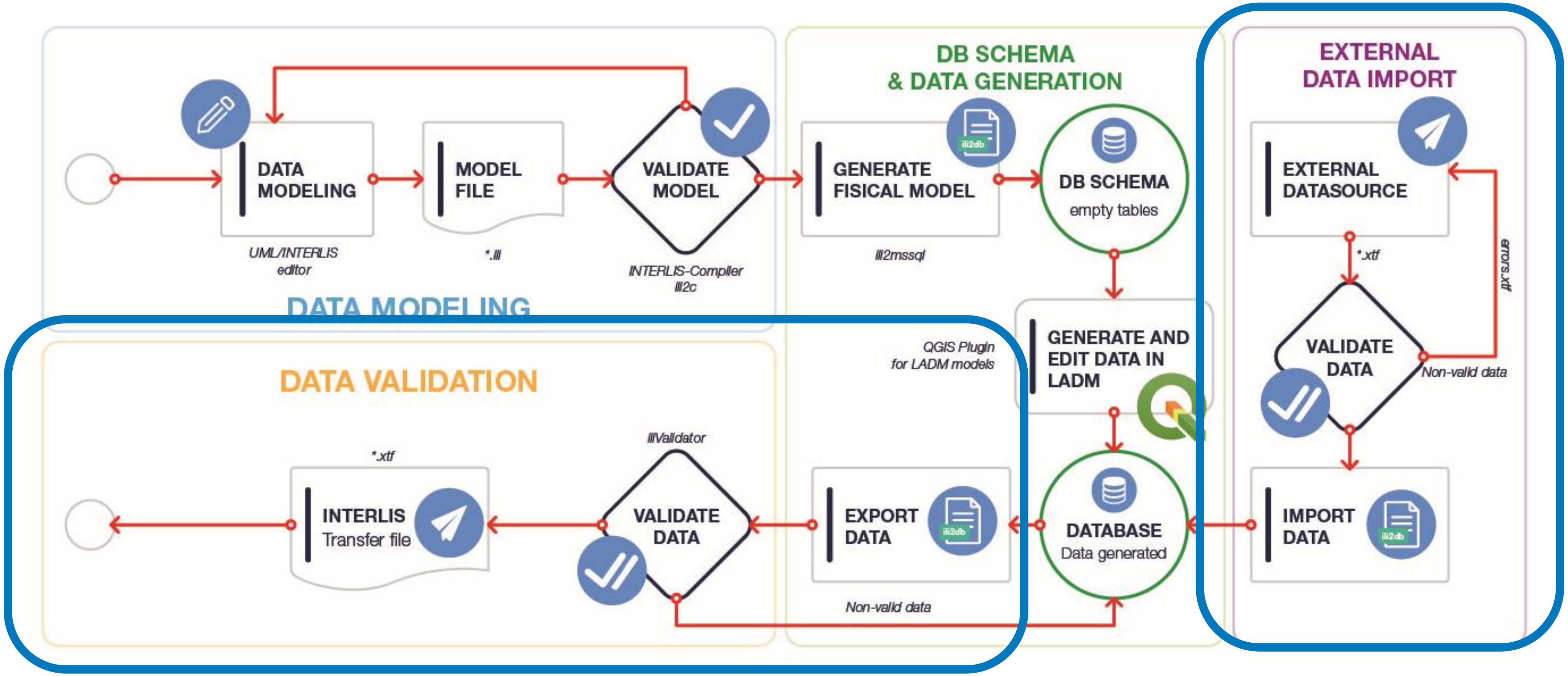
https://www.interlis.ch/download/interlis2/ili2-refman_2006-04-13_e.pdf

https://drive.infomaniak.com/app/share/189474/c1d19a50-43c8-4b34-b238-d4f7814f37c7

LandNetwork

Import / Export and Data Validation

Typical INTERLIS implementation Workflow



Mejía et al., 2017

LandNetwork

Ili2db custom attributes

- --t_id_Name *columnName*

  defines the name of the internal/technical id. Default : t_Id

INTERLIS Definition

```
CLASS A =
END A;
```

SQL Result

```
CREATE TABLE A (
    columnName integer PRIMARY KEY
);
```

- --createTidCol

  Creates an additional t_ili_tid

INTERLIS Definition

```
CLASS A =
END A;
```

SQL Result

```
CREATE TABLE A (
    T_Id integer PRIMARY KEY,
    T_Ili_Tid varchar(200) NOT NULL
);
```

Ili2db custom attributes

- --importTid

  reads the TID from the transfer file into an additional column t_ili_Tid (otherwise only classes with a stable OID gets this column)
- --exportTid

  When exporting uses the value of the t_ili_Tid as transfer identification (TID in the xtf file), Requires a schema created using **--createTidCol**

- --createBasketCol

  creates in every table an additional column T_basket (that is an FK to t_ili2db_basket) to identify the basket of this record

INTERLIS Definition

```
CLASS A =
END A;
```

SQL Result

```
CREATE TABLE A (
    T_Id integer PRIMARY KEY,
    T_basket integer NOT NULL
);
```

LandNetwork

Ili2db custom attributes

- --createDatasetCol

  creates in every table an additional column T_datasetname to identify the dataset of this basket. Requires also the option --dataset. The column is redundant to column datasetname in table t_ili2db_dataset

INTERLIS Definition

```
CLASS A =
END A;
```

SQL Result

```
CREATE TABLE A (
   T_Id integer PRIMARY KEY,
   T_datasetname varchar(200) NOT NULL
);
```

- --import, --replace, --delete, --export & **--dataset *datasetname***

  Imports, replace, delete, or export all records related to a given dataset, **requires --createBasketCol.**

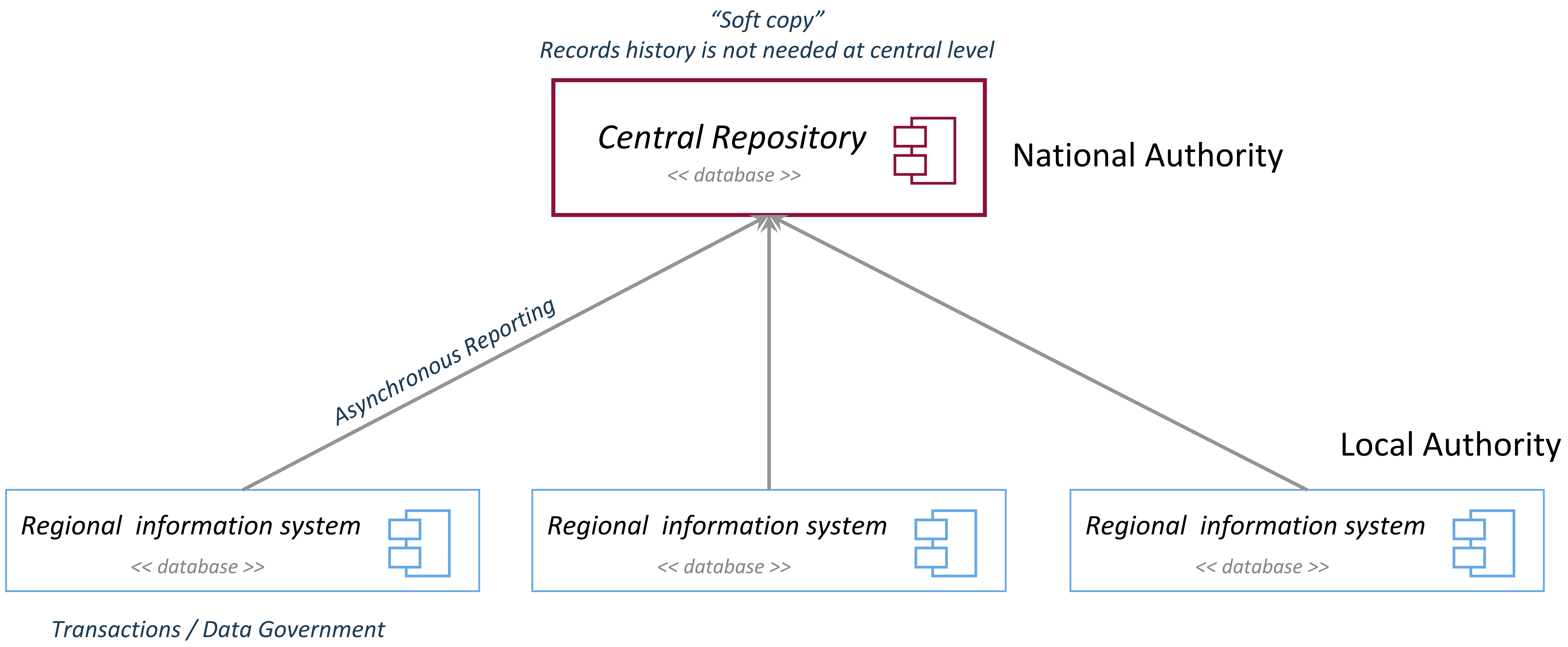
- --baskets BID Id of the baskets to be exported, imported or replaced

LandNetwork

LandNetwork

INTERLIS Baskets

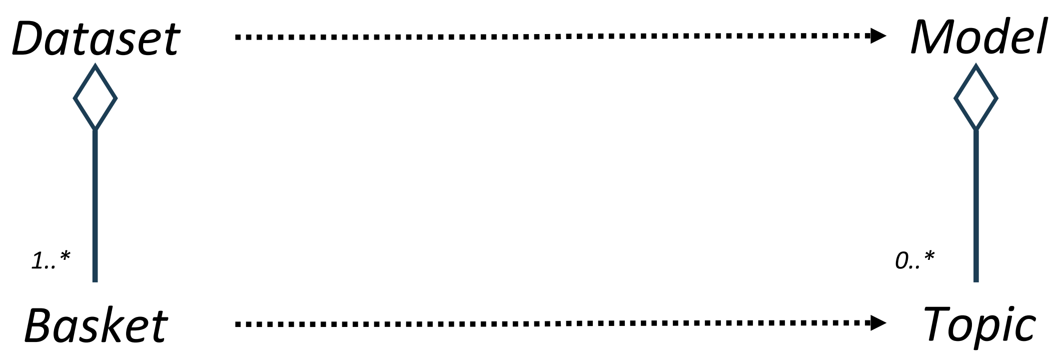
Distributed Information Management


"Soft copy"
Records history is not needed at central level
Central Repository
<< database >>
National Authority
Asynchronous Reporting
Local Authority
Regional information system
<< database >>
Regional information system
<< database >>
Regional information system
<< database >>
Transactions / Data Government

LandNetwork

Baskets and Datasets

“Minimum collection of related objects”



- Basket contains all objects of a geographic area (e.g., a city) or a domain (e.g., water utilities)
- All objects inside a basked are transferred and checked as a unit. (minimum transfer unit)
- Objects inside a basket should be responsibility of a single authority.

INTERLIS Constraints

LandNetwork

Constraints

- Rules that must be followed by a combination of values or certain objects
- Are part of the definition of Classes, Associations or Views

Mandatory

Applies to all objects of the class, constraint is evaluated for each one.

```
MandatoryConstraint = 'MANDATORY' 'CONSTRAINT'
                                  Logical-Expression ';'
```

```
CLASS Employee =
  firstName : MANDATORY TEXT*30;
  isRetired : BOOLEAN;
  salary : 0..250000 [CHF];
  MANDATORY CONSTRAINT
    !! If employee is not yet retired,
    !! a salary must be defined
    isRetired==#true OR DEFINED(salary);
END Employee;
```

Constraints

Unique

Describes a combination of attributes that must be unique across all object on the basket

```
UniquenessConstraint = 'UNIQUE’ GlobalUniqueness ';’.

GlobalUniqueness = UniqueEl.

UniqueEl = ObjectOrAttributePath { ',’
                   ObjectOrAttributePath }
```

```
CLASS Employee =
   firstName : TEXT*30;
   lastName : TEXT*30;
   taxIdentificator: Mandatory TEXT*10;
   !! All Employees have a unique Tax ID
   !! All employees have a unique name
   UNIQUE taxIdentificator;
   UNIQUE firstName, lastName;
END Employee;
```

Constraints

Set

Used to describe more complex rules that must be followed for a group (set) of object inside the basket

```
SetConstraint = 'SET' 'CONSTRAINT’
                        Logical-Expression ';'.
```

```
CLASS Parcel =
   type: (private, public, ….);
   Geometry: SURFACE ...;
   SET CONSTRAINT WHERE type = #public :
      areAreas(ALL, UNDEFINED, >> Geometry);
END Parcel;
```

All public parcels (type = Public) should be an aggrupation without overlaps and gaps (Tessellation)

LandNetwork

Functions

- Used in constraints to add logic to validations
- Usable to return complex calculations

Custom function workflow:

Define the function -> use the function -> implement the function (external tools)

Predefined functions Use qualified name (INTERLIS.len())

Text Functions

- FUNCTION len (TextVal: TEXT): NUMERIC;
- FUNCTION lenM (TextVal: MTEXT): NUMERIC;
- FUNCTION trim (TextVal: TEXT): TEXT;
- FUNCTION trimM (TextVal: MTEXT): MTEXT;

Element count Functions

- FUNCTION elementCount (bag: BAG OF ANYSTRUCTURE): NUMERIC;
- FUNCTION objectCount (Objects: OBJECTS OF ANYCLASS): NUMERIC;

Functions

Check Type Functions

- FUNCTION myClass (Object: ANYSTRUCTURE): STRUCTURE;
- FUNCTION isSubClass (potSubClass: STRUCTURE; potSuperClass: STRUCTURE): BOOLEAN;
- FUNCTION isOfClass (Object: ANYSTRUCTURE; Class: STRUCTURE): BOOLEAN;
- FUNCTION isEnumSubVal (SubVal: ENUMTREEVAL; NodeVal: ENUMTREEVAL): BOOLEAN;
- FUNCTION inEnumRange (Enum: ENUMVAL; MinVal: ENUMTREEVAL; MaxVal: ENUMTREEVAL): BOOLEAN;

Misc. Functions

- FUNCTION convertUnit (from: NUMERIC): NUMERIC;
- FUNCTION areAreas (Objects: OBJECTS OF ANYCLASS; SurfaceBag: ATTRIBUTE OF @ Objects RESTRICTION (BAG OF ANYSTRUCTURE); SurfaceAttr: ATTRIBUTE OF @ SurfaceBag RESTRICTION (SURFACE)): BOOLEAN;

LandNetwork

iliValidator

iliValidator

Check configurations

- --config filename

  Name of file to config validation (same file as for ilivalidator)
- --disableAreaValidation

  Avoid checking AREA topology rules
- --disableConstraintValidation

  Deactivates the constraint check.
- --forceTypeValidation

  only "multiplicity" can be relaxed in config file
- --log filename

  write log messages to the given file
- --xtflog filename

  write log messages as according to IliVErrors model (Can be read with ili2pg to visualize the validation errors)

iliValidator Configuration

- Configurations can be set inline in the interlis model .ili file using meta-attributes !!@ ilivalid.
- Can be done in a TOML File (additional to the interlis model)
- Main Functions:
  - Decreasing level of validation “relaxing rules”
  - Creating custom validation messages for falling constraints

TOML File structure

```
# global section
["PARAMETER"]
additionalModels=“ModelNamesSplitBySemicolon“

# interlis model element name (only once)
[" UtilityNetworkModel.WaterNetworkTopic.PipelineClass.Kind"]
multiplicity="warning"
```

LandNetwork

Decrease Level of validations

Attributes validation

```
["UtilityNetworkModel.WaterNetworkTopic.PipelineClass.Kind"]
# validate multiplicity
# if MANDATORY: validate if there is a value
# if BAG/LIST: validate if right number of elements
# Possible values: on/warning/off
multiplicity="warning“
# inline definition:  !!@ilivalid.multiplicity=warning


# validate type, e.g., if a number is in range
# Possible values: on/warning/off
type="off“
# inline definition:  !!@ilivalid.type=off
```

Decrease Level of validations

Constraints validation

```
[" UtilityNetworkModel.WaterNetworkTopic.PipelineClass.ConstraintName"]
# validate constraint
# Possible values: on/warning/off
check="warning“
# inline definition:  !!@ ilivalid.check=warning


#message text if the constraint fails
# attribute values can be added using {}
msg= “Pipeline size not belong to {kind} type"
msg_fr="La taille du pipeline n'appartient pas au type {kind}“

# inline definition:  !!@ ilivalid. msg_fr = “La taille du pipeline n'appartient pas au type {kind}"
```

LandNetwork

Decrease Level of validations

Association validation

```
MODEL Example =
   TOPIC SchoolMgmt =
      CLASS School =
      END School;
      CLASS Person =
      END Person;
      ASSOCIATION person2school=
         primarySchool -- {0..*} School;
         director -- {1} Person;
      END;
END Example.
```

```
[" Example. SchoolMgmt. Person2school.director"]
# validate multiplicity/number of associated objects
# Possible values: on/warning/off
multiplicity="warning“

# validate type of target object
# e.g., if referenced director is a Person object
# Possible values: on/warning/off
type="off"
```

Model with additional constraints

Using views to add constraints

```
MODEL AdditionalConstraints =
   IMPORTS ExampleModel;

   VIEW MyViewOfB
      PROJECTION OF ExampleModel.ClassB;
      =
         ALL OF ClassB;

         !!@ilivalid.check=warning
         !!@ilivalid.msg= “an important message"
         MANDATORY CONSTRAINT …;
   END VB;
END AdditionalConstraints.
```

LandNetwork

Pending Questions

Questions during the course

- Is possible to get the sql script used to create a database?

  Ussing *--createscript filename* when running –schemaimport will create an additional sql script. (it also creates the table schema, it's not possible to just create the script)

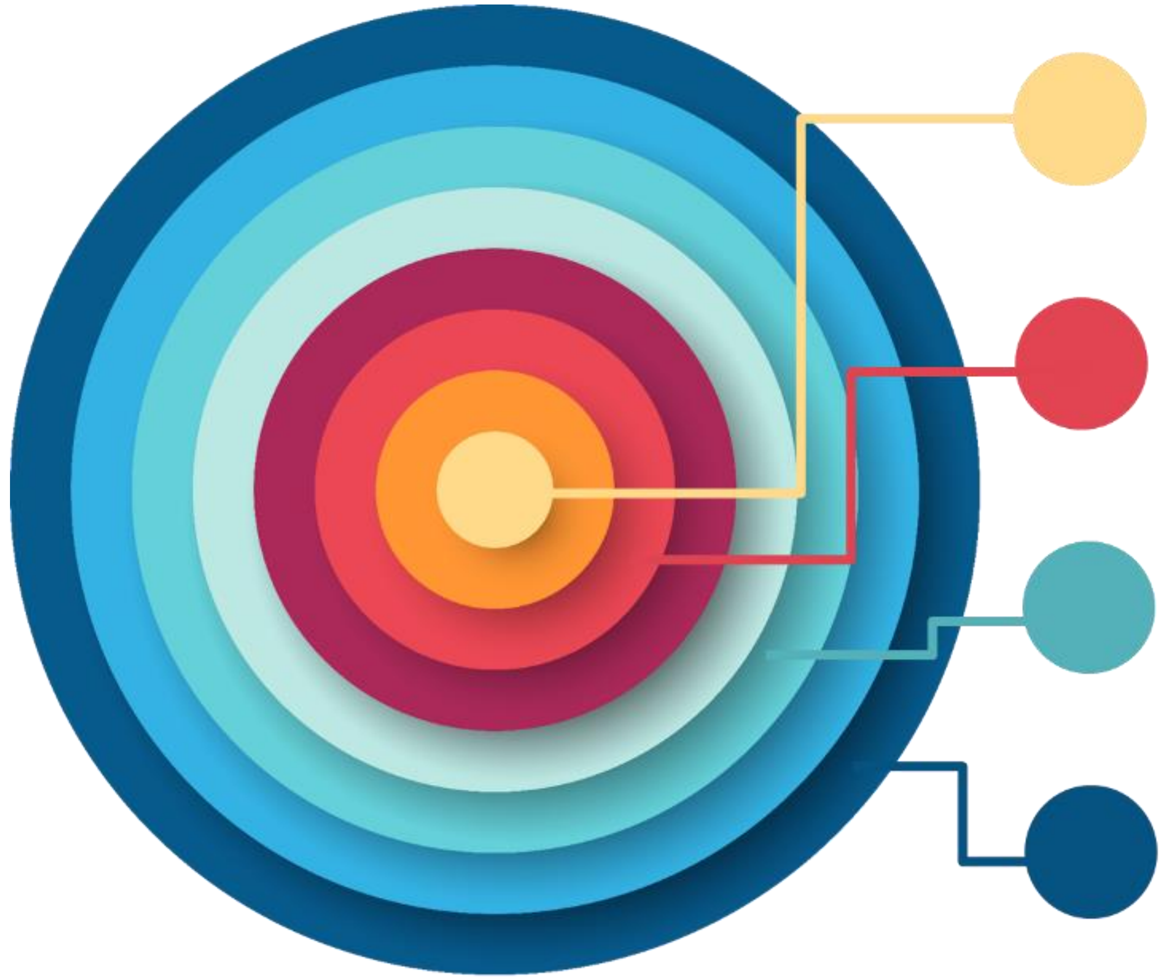
- Are any other implementation of INTERLIS tools beyond ili2 family

  Additional to *Eisenhut Informatik's* tools (ili2db, ili2c, iliValidator) other tools/libraries used to work with INTERLIS:

  INTERLIS Studio, developed by GeoCom AG (few information and not-known implementations), and

  INTERLIS Tools, iG/Check, GeoShop, created by infoGrips GmbH distributed under WORLD WIDE SINGLE USER LICENSE for commercial and non-commercial purposes, also has an online validation service (paid service)

Real World modelling examples – Colombian LADM profile



ISO 19152 – LADM
Basic definitions on Land Administration

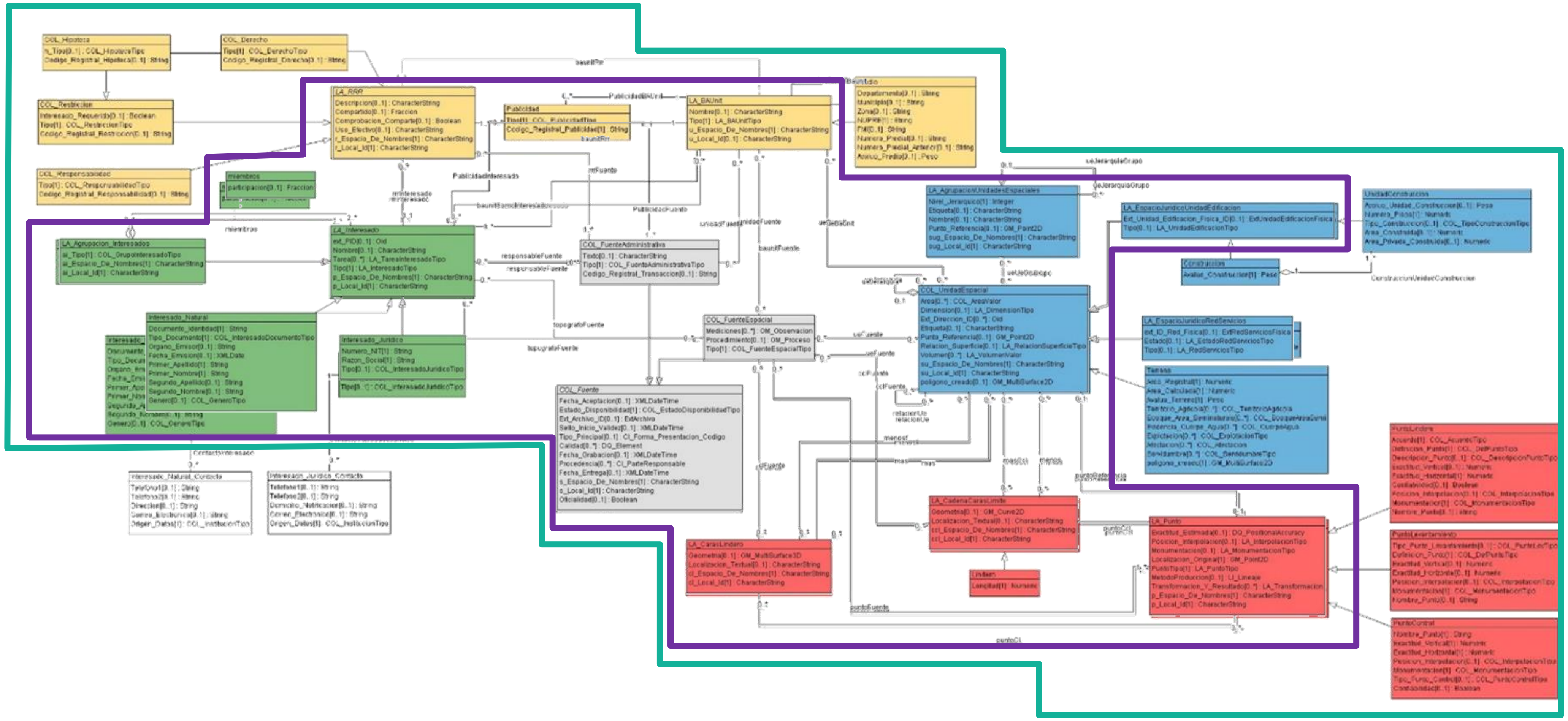
LADM-Col model
Core model for Colombian Land Administration

National Cadaster Model
Basic model for al cadastral agencies at local level

Application models
Extended topics and custom classes for local administration purposes

LandNetwork

Real World modelling examples – Colombian LADM profile

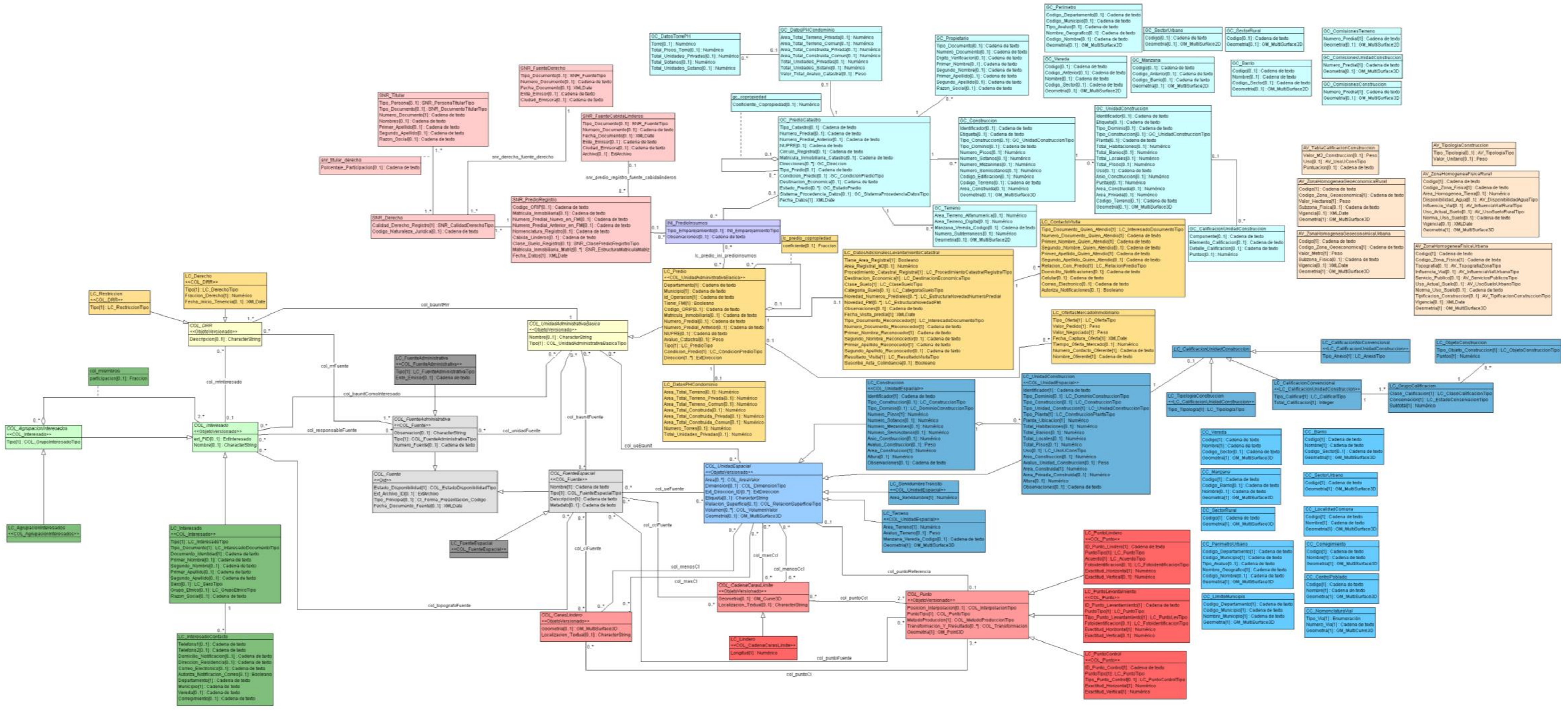


Colombian LADM country profile

National Cadaster Model

Sub-National cadaster Agency Model

Includes valuation and auxiliary surveying classes (only needed at local level)



LandNetwork

Thank you

Fabian.mejia@landnetwork.ch