



**ELF** EUROPEAN  
LOCATION  
FRAMEWORK

# Challenges and potential solutions to implement temporal aspects in INSPIRE specifications

**Presentation to:** INSPIRE Conference 2015, Lisbon

**Author:** Dominique Laurent

**Date:** 26<sup>th</sup> May 2015

ISN.15.064

# Agenda

- ★ The Basemap idea
- ★ The Basemap model
- ★ Portrayal
- ★ How does it may look like?
- ★ Conclusions

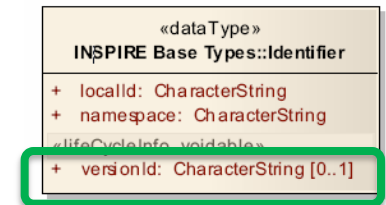
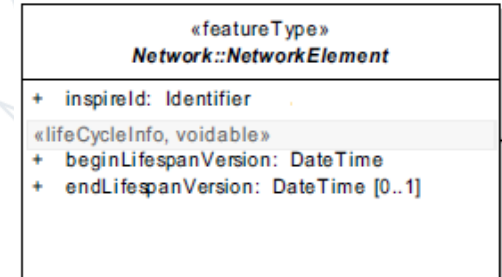
# Theory: INSPIRE mechanism for incremental updates

## ★ Most of INSPIRE data models include:

- ★ Inspire identifier
  - Unique
  - **persistent**
- ★ Temporal attributes related to the data base life-cycle
  - beginLifespanVersion
  - endLifespanVersion



Versioning of objects



## ★ Users can get incremental updates (e.g. evolutions between $t_1$ and $t_2$ ) just by querying on temporal attributes

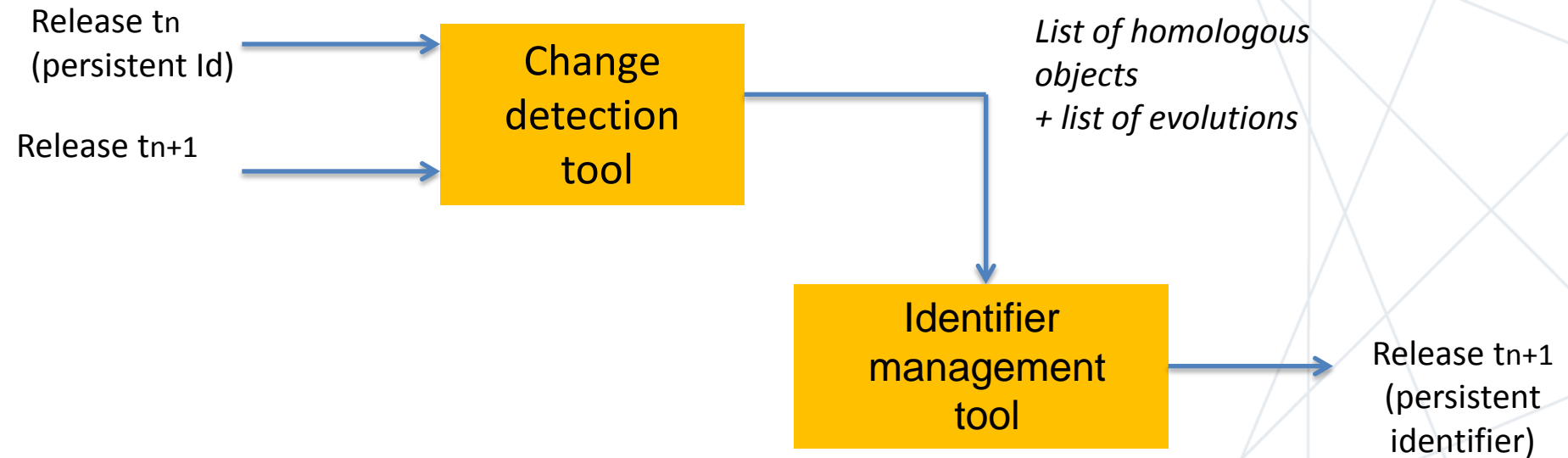
- ★ beginLifespanVersion after  $t_1$  and before  $t_2 \Rightarrow$  to get new (created) objects or new versions of modified objects
- ★ endLifespanVersion after  $t_1$  and before  $t_2 \Rightarrow$  to get old (deleted) objects or old versions of modified objects

# Practice: many issues

- ★ Many data producers deliver only valid data (by regular releases) but do not give access to historical data
  - ★ => Users can't get information about old /deleted objets
  
- ★ Not always persistent identifiers
  - ★ Persistent identifiers missing in source data
  - ★ Persistent identifiers lost during the transformation process
    - Split features
    - Merge features
  
- ★ Temporal attributes
  - ★ May be missing in source data
  - ★ May be no longer reliable due to transformation process
    - More data in source than in INSPIRE
      - » Overdetections
    - Main source data + ancillary data to fill INSPIRE (e.g. by joining tables)
      - » underdetections

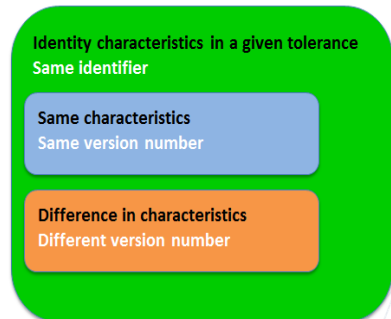
# The ELF proposal: the change detection tool

- ★ Principle: ensure **persistent** identifiers and provide temporal attributes, by comparing objects coming from 2 releases at  $t_n$  and  $t_{n+1}$



# Persistent identifier rules

- ★ The inspire identifier is about the database feature not about the real-world entity
- ★ General issue:
  - ★ When is an object considered as modified (same identifier, new version)?
  - ★ When it is considered as a new object (new identifier)?
- ★ An object is defined by a set of properties (geometry, semantic, ...)
  - ★ Significant change in main properties => new object => new identifier
    - **Which are the main properties?**
    - How big the change should be?
  - ★ Any other change => new version



# Persistent identifier rules: state-of-play

## ★ INSPIRE context

- ★ No common rules
  - Life-cycle rules up to each data provider
  - Some good practice examples in the Methodology (guidelines for harmonisation)

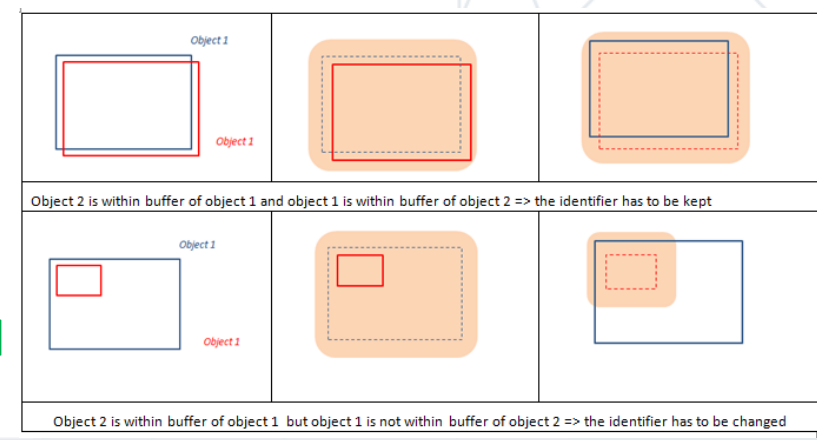
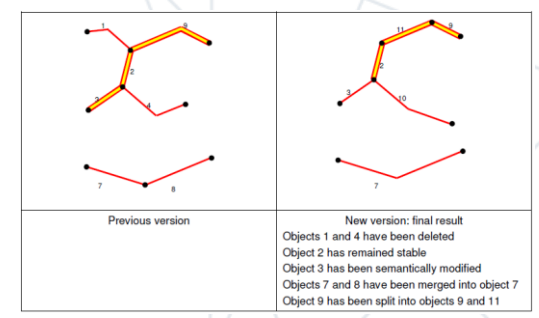
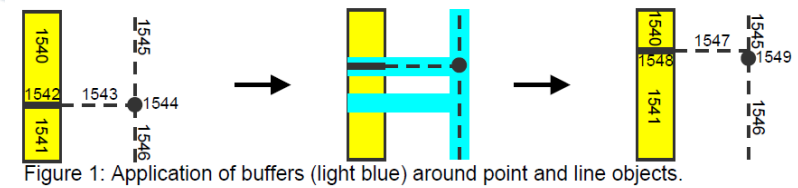
## ★ ESDIN (ELF predecessor) project:

- ★ Proposal for some common life-cycle rules for "simple" objects

## ★ Current IGN change detection tool

- ★ Rules for "simple" objects

★ => in all existing rules, **geometry is considered as main property, as identity property**



# Persistent identifier rules: new challenge

- ★ Existing life-cycle rules considered **geometry as identity characteristic for spatial objects**
  
- ★ But not in INSPIRE data models, situation is more complex:
  - ★ Objects with multiple geometries (AD, BU, CP, ...)
  - ★ Objects with generic geometry (GM\_Object, GM\_Primitive)
  - ★ Objects with indirect geometries
    - Aggregate objects : set of other objects (Road, WatercourseLinkSequence, ...)
    - Properties in theme TN: attached by linear referencing to geometric objects
  - ★ Objects with no geometry at all (AD components)
  - ★ ...
  
- ★ => **new rules** are required for these cases



# Persistent identifier: ELF rules

## ★ General rule:

One of the identity characteristics of an object is its **classification**: in general, this classification is given by the **feature type**

*NOTE : In some specific cases, the feature type may be completed by a classification attribute (e.g. generic feature type such as NamedPlace, GovernmentalService).*

★ Existing life-cycle rules considered **geometry as identity characteristic for spatial objects**

★ But not in INSPIRE data models, situation is more complex: