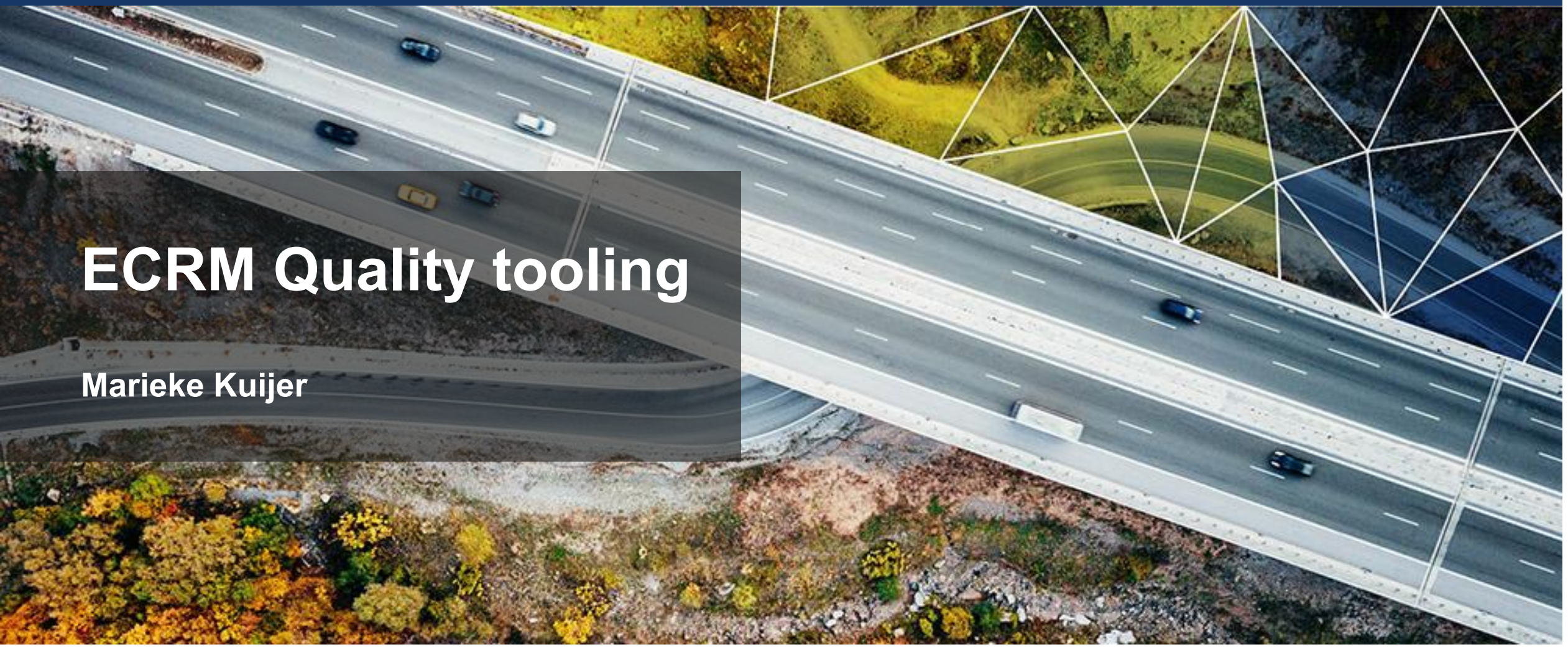


ECRM Quality tooling

Marieke Kuijer

QKEN - 26 November 2026

Introduction



Marieke Kuijer

- Geodata Specialist - Kadaster NL
- Involved with EuroGeographics since 2017
  - Regional Coordinator for ERM Group A
  - ERM Quality tooling
  - ECRM Quality tooling

ERM = EuroRegionalMap (medium scale)

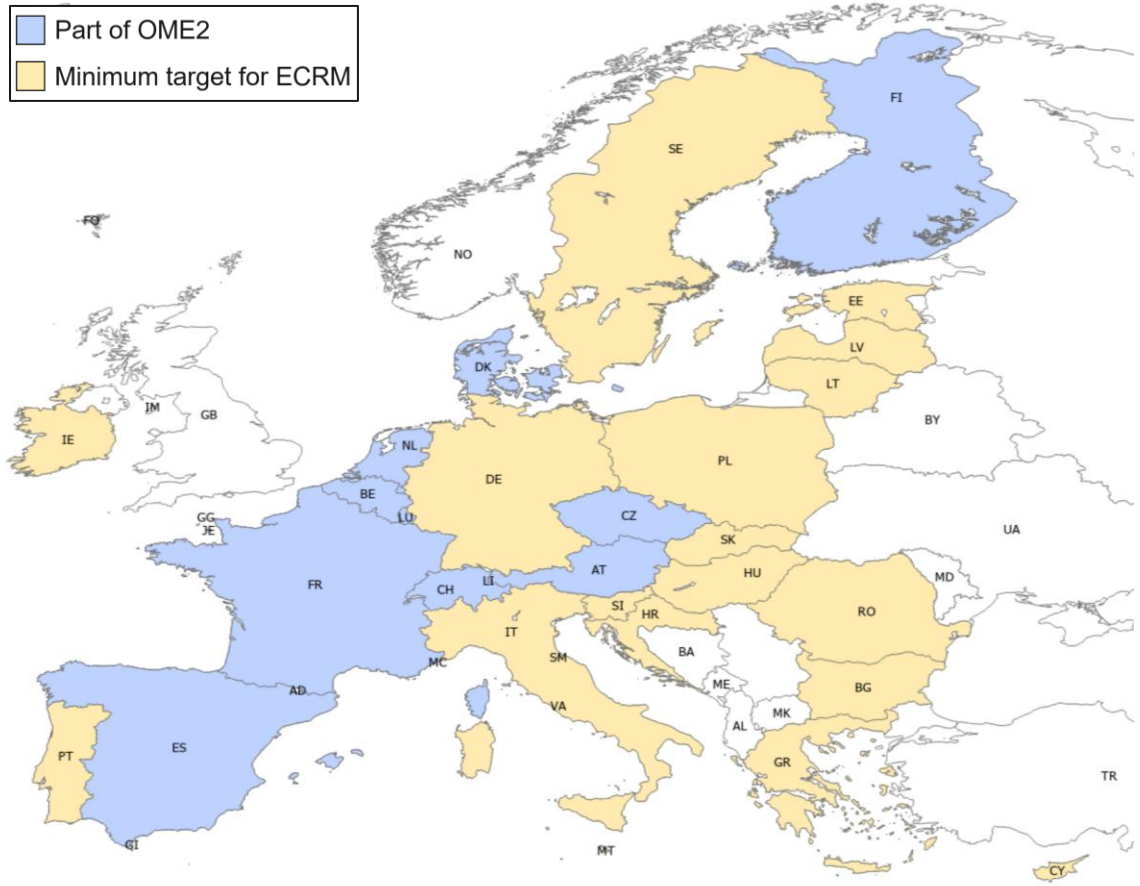
ECRM = EuroCoreReferenceMap (large scale)

Copyright © 2023 EuroGeographics

ECRM at a glance

Euro Core Reference Map (ECRM): Creating a new authoritative foundation for cross-border applications in Europe

- **Pan-European coverage (EU27 + X)**: harmonised and edge-matched data across national borders
- **Large-scale (~1:10 000)**: consistent high-resolution topographic information
- **Authoritative data**: trusted official sources from national mapping agencies
- **3 themes:** Admin, Hydro and Trans



Copyright © 2026 EuroGeographics

Project Consortium

EuroGeographics

- Project Management and coordination
- Stakeholder engagement
- Communication and Dissemination

Federal Agency for Cartography and Geodesy

Federal Agency for Cartography and Geodesy

- Technical on-boarding of data producers
- Data Production and release management

Institut national de l'information géographique et forestière

- Technical coordination
- Maintenance and development of harmonisation and edge-matching tools



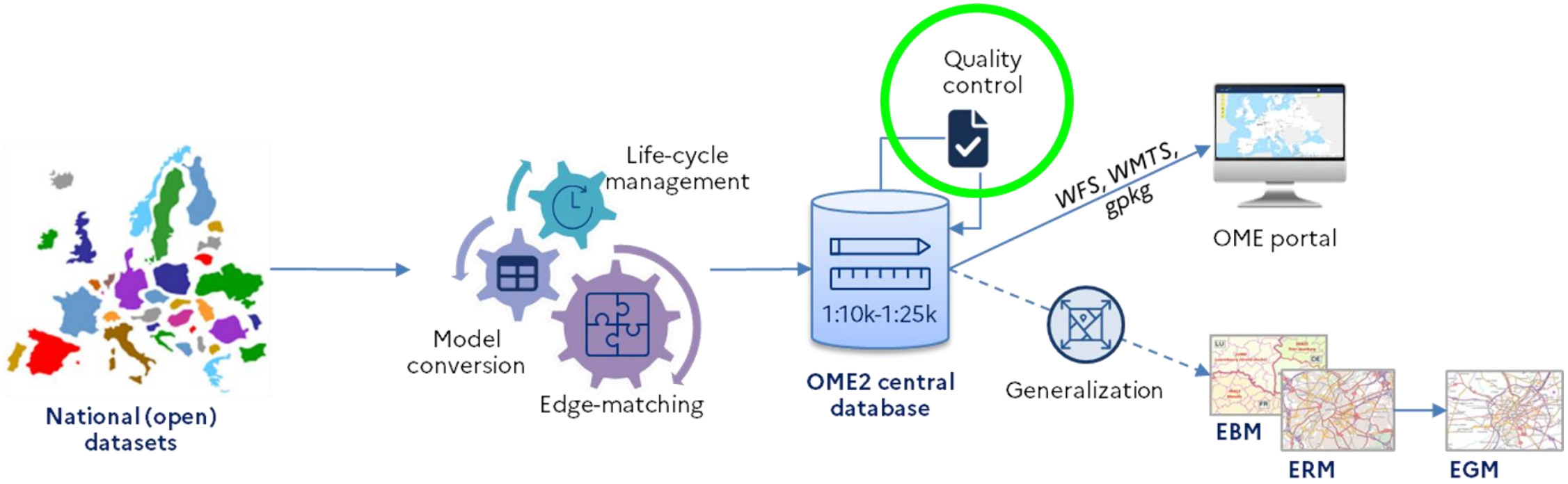
2026 - 2028

Kadaster Netherlands

- Maintenance and development of quality and generalisation tools

Copyright © 2026 EuroGeographics

Overview of the ECRM production process



Copyright © 2026 EuroGeographics

Quality control tool

Model conversion

Edge matching

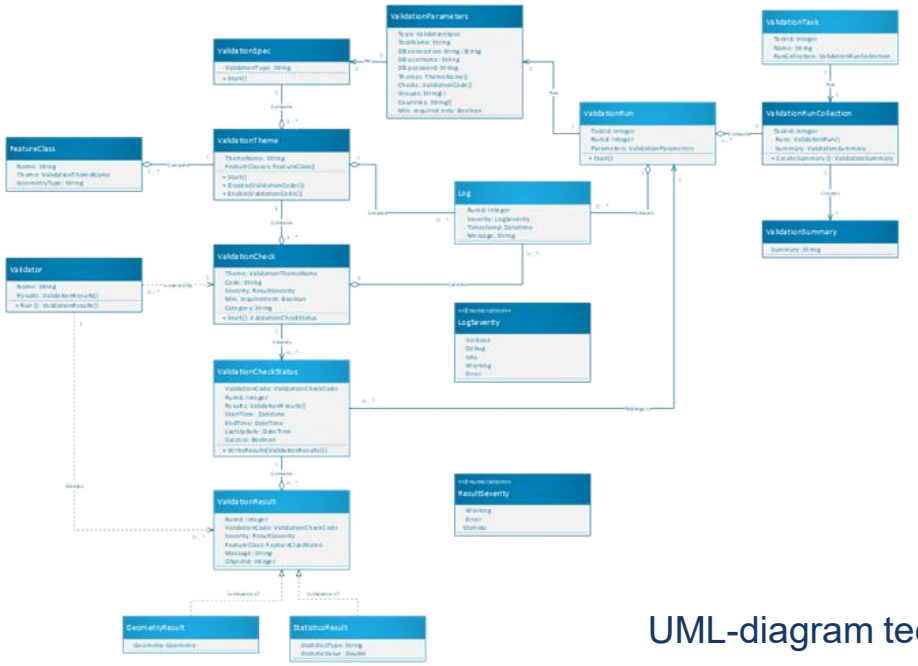
Life-cycle management

Quality control

Update procedure

Generalization

Objective: Set up and maintain tooling that checks the quality of the ECRM datasets



UML-diagram technical design

Source code: internal (on GitHub later)

Language: Python

Status: functional

Main prerequisites:

- In IGNF cloud environment
- Opensource
- Flexibel & re-usable

Copyright © 2023 EuroGeographics

Flowchart



Input database ECRM

Args file

Quality tool

Data Validation checks
Quality Assessment checks

3 themes:
Administrative Units
Transportation network
Hydrography network

docker

pyQGIS

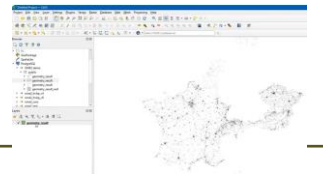
Results database

Summary report.txt

Log file.txt

Visual inspection of results

QGIS



MVP Dashboard

Copyright © 2023 EuroGeographics

eurogeographics

Validators/

- Modeled after ERM Validator
- Small tools that perform a particular validation check, e.g., MinimumAreaValidator

```
class MinimumAreaValidator(FeatureValidator):
  @classmethod
  def validate(cls, run_id, validation_code,
               severity: str, feature_classes, minimum_area) :

    results: list[ValidationResult] = []

    # ...
    d = QgsDistanceArea()
    for feature_class in feature_classes:

       # ...
       for feature in feature_class.getFeatures():

         # ...
         feature_area = d.measureArea(feature.geometry())
         if  feature_area < minimum_area:
           msg = f"""Feature with objectid '{feature['objectid']}'
                     has an area of {round(feature_area, 1)}m2
                     which is smaller than the minimum area of
                     {minimum_area}m2."""

           country = cls.get_attribute(feature, 'country')
           result = cls.create_result(
             run_id,
             validation_code,
             severity,
             feature_class,
             feature,
             message,
             country
           )
           results.append(result)

       return results
```

Validation_specs/

- ***Declarative*** encoding of validation **specs** *consisting of* **feature classes** and **validation rules**

```
# Example snippet validation_specs/dv1/transport.py

class ferry_crossing(feature):
    objectid: uuid[primary_key, notnull]
    country: varchar[length(8)]
    begin_lifespan_version: timestamp
    end_lifespan_version: timestamp
    w_national_identifier: varchar[length(255)]
    xy_source: varchar[length(255)]
    z_source: varchar[length(255)]
    w_scale: varchar[length(80)]
    w_release: int4
    geom: MultiLineStringZ[srid(3035)]
    name: jsonb
    label: varchar[length(255)]
    ferry_use: varchar[length(255)]
    tent_network: varchar[length(255)]
    access_restriction: varchar[length(255)]


T006f = ferry_crossing.LengthMustBeAtLeast(100,
            check_multilines_per_linestring=False
        ).TreatAsWarning()
```

Some example questions

Data structure
- Does the data match the ECRM data model?
- Are all feature classes present in the dataset?
- Is all data in the correct coordinate system?
- Are all identifiers unique?

Geometry
- Are there invalid geometries?
- Are all geometries within the specified extent?
- Are multipart geometries used correctly?
- Are there self-intersections?
- Are there empty or duplicate features?
- Are all line features longer than the minimum length?

Topology
- Are there any gaps or overlaps between adjacent polygons?
- Are there any disconnections within the hydrography and transport network?

Edge matching
- Do the transport and hydrography networks connect seamlessly across border areas?

Quality assesment
- What is the feature count per feature class?
- What is the completeness rate of mandatory attribute values?
- What percentage of features have missing or empty values for specific attributes?

Copyright © 2024 EuroGeographics

Screenshot of the code

validation_specs\dv1\trans.py

```
begin_theme('TRANS', 'tn')

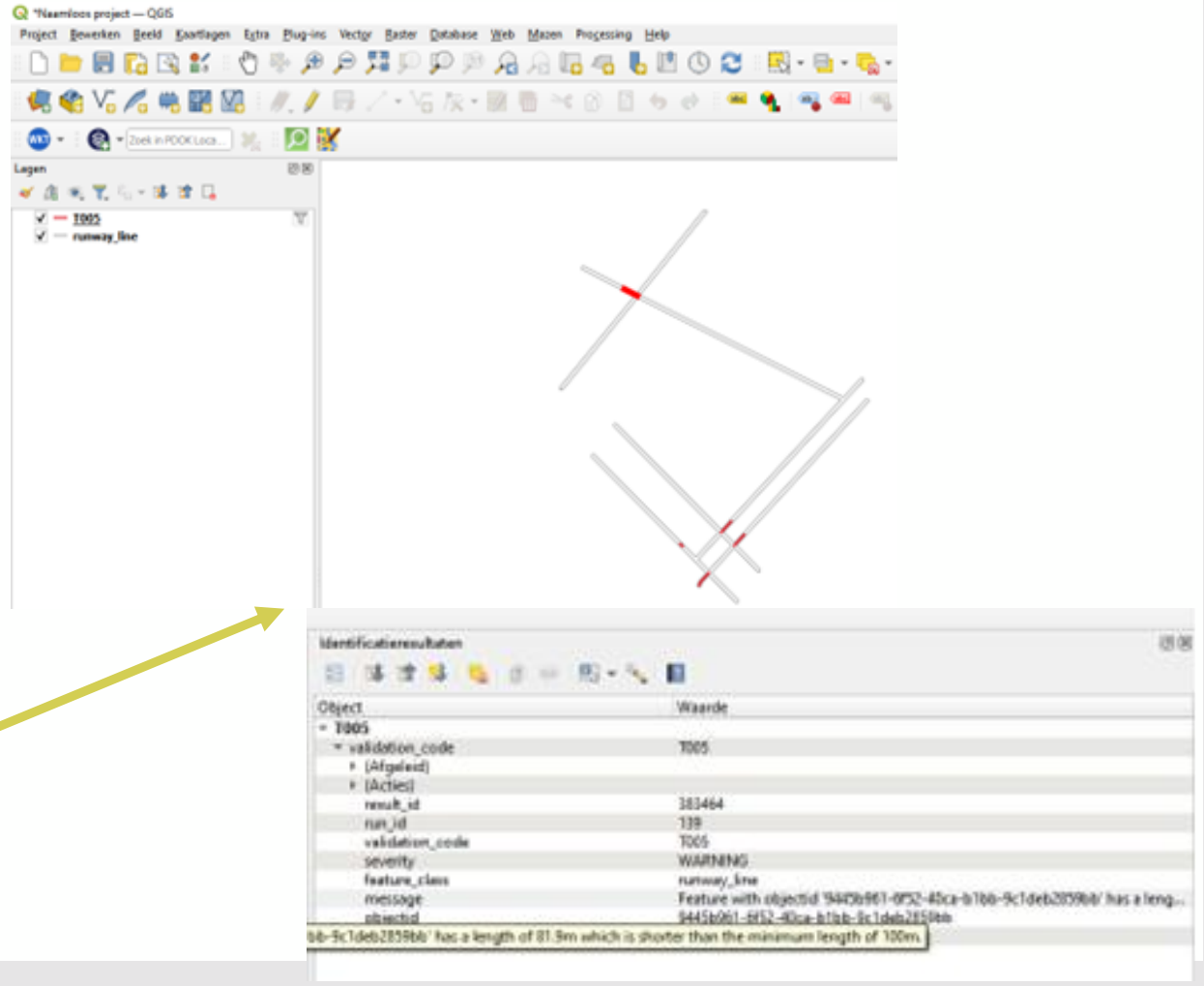
class runway_line(feature):
    objectid: uuid[notnull]
    country: varchar[length(8)]
    begin_lifespan_version: timestamp
    end_lifespan_version: timestamp
    geom: multilinestringz[srid(3035)]
    surface_composition: varchar[length(255)]
    w_national_identifier: varchar[length(255)]
    w_step: integer
    xy_source: varchar[length(255)]
    z_source: varchar[length(255)]
    w_scale: varchar[length(80)]
    w_release: integer


# General content
T001 = runway_line.MustComplyWithDataschema()

# Geometric resolution
T002 = runway_line.CrsMustBe(3035)
T003 = runway_line.MustHaveValidGeometry()
T004 = runway_line.MustHaveGeometryType(MultiLineStringZ)
T005 = runway_line.LengthMustBeAtLeast(100, check_multilines_per_linestring=False).TreatAsWarning()

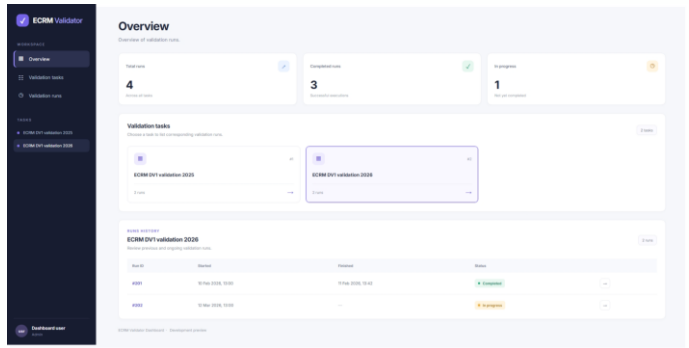
# Data model and attribute resolution
T010 = runway_line.objectid.MustNotBeNull()
T020 = runway_line.surface_composition.MustBeOfValues(['asphalt','concrete','grass','void_unk'])

end_theme()
```



Next steps – end 2028

- Improve current tooling
- Add checks for more scales/countries/(themes)
- Deploy on IGNF infrastructure
- Establish QA process
- Build MVP dashboard(s) to visualize the results



Copyright © 2018 EuroGeographics

Questions or suggestions?

Contact:

marieke.kuijer@kadaster.nl
tony.baving@kadaster.nl

Copyright © 2026 EuroGeographics