Automatic generalisation from Large to Medium scale (Task 5.1)

Advisory report on the feasibility of using open-source tools to generalise EBM and ERM data from the High-value Large-scale dataset (HVLSD).

File name			
Version	Author	Date	Comments
1.0	Tony BAVING	21 November 2025	First version
1.1	Konan PRUIKSMA	10 December 2025	Review
1.2	Marieke KUIJER	15 December 2025	Review

OME2 Co-funded by the European Union

Digital Europe Programme
Grant Agreement No 101100625

Contents

1. Introduction

Kadaster Netherlands has a long and distinctive history in the field of automated cartographic generalisation. Well before the start of the OME2 project, Kadaster developed a fully automated generalisation workflow for national topographic data. At that time, this work was carried out using ESRI-based technology, making Kadaster one of the first national mapping agencies in Europe to successfully implement automated generalisation in an operational environment.

This national generalisation framework, designed to derive medium- and small-scale map products from the large-scale TOP10NL dataset, delivered significant operational insights and efficiency gains. The automated workflow consisted of dozens of algorithmic processes and resulted in a major acceleration of the production cycle:

- Previously, updating the entire country required 26 cartographers working for approximately five years, a total of about 130 man-years of effort.
- With automation, a complete national update could be produced in approximately two weeks.

This breakthrough allowed the Netherlands to be generalised consistently, quickly and fully automatically, without manual post-editing. The experience gained in this period, both technically and organisationally, provides an important knowledge base for approaching generalisation challenges at an European scale.

With the start of the OME2 project, a deliberate shift was made: the development and evaluation of new generalisation tools using exclusively open-source technologies. This aligns with the core principles of OME2, including reusability, transparency and interoperability within the evolving European data space. Within Work Package 5, and specifically Task 5.1 and the Deliverable 5.3, Kadaster Netherlands focused on:

1. Investigating open-source generalisation tools for deriving scales 1:50k, 1:100k and 1:250k from the High-value Large-scale Prototype (HVLSP); and
2. Developing a prototype to assess the feasibility of automatically generating two pan-European reference products:
   - EuroBoundaryMap (EBM / BND)
   - EuroRegionalMap (ERM / TRANS)

This report presents the approach, results and findings obtained within OME2 using a fully open-source workflow.

The following chapters describe which components were successfully achieved, where limitations were encountered, and what lessons can be drawn for future development of pan-European generalisation pipelines. In this way, the report contributes to the broader goal of establishing sustainable and harmonised generalisation processes across Europe within OME2 and successor programs.

2. Objectives and Scope

The objective of Kadaster's contribution within OME2 is defined as the investigation, assessment and prototyping of generalisation tools intended to support the production of harmonised mid-scale and small scale datasets at the European level. Within Task 5.1 (T5.1), the scope includes evaluating whether the newly developed large scale harmonised dataset can serve as an appropriate input source for the automated derivation of EuroBoundaryMap (EBM) and EuroRegionalMap (ERM).

The assignment comprises the following components:

1. Investigation of generalisation tools and methods

This includes the examination of open source generalisation techniques capable of transforming large scale input data to smaller scale representations (1:50k, 1:100k and 1:250k). The assessment covers geometric simplification, topological preservation, feature aggregation and rule based transformations relevant to harmonised European datasets.

2. Prototype development for EBM, ERM and thematic generalisation

The task requires the creation of a prototype workflow to evaluate the feasibility of deriving the administrative units dataset (AU), the transport dataset (TRANS) and the hydrographic dataset (HYDRO) from the pan European large scale data model using exclusively open source tools. The prototype serves as a feasibility assessment rather than a production workflow.

3. Assessment of maturity, limitations and future requirements

This includes identifying constraints related to data heterogeneity, algorithmic performance, cross border continuity and the practical applicability of open source tools in a multi country environment.

The geographical areas of interest covered in this assignment are the Netherlands, Belgium and France. For each of these areas, three thematic domains are included in scope: Administrative Units (AU), Transport (TRANS) and Hydrography (HYDRO).

Activities performed outside the OME2 project, including earlier ESRI based generalisation processes, are explicitly considered out of scope. The work carried out within OME2 is limited to open source technologies and focuses on research, experimentation and feasibility assessment rather than operational production.

3. Context within OME2 and Work Package 5

The Open Maps for Europe 2 (OME2) project builds on the success of the original Open Maps for Europe initiative by creating a new production system for harmonised, high-value large-scale pan-European datasets (HVLSD). These datasets—covering administrative units, transport networks, and hydrography—are essential for supporting the objectives of the EU Open Data Directive and enabling the reuse of authoritative public sector geospatial data across Europe.

Within this framework, Work Package 5 (WP5) delivers the technical tools required to achieve harmonisation and interoperability of national datasets. WP5 focuses on the development and maintenance of:

- Conversion tools to transform national data models into a common pan European specification, including harmonisation of geometrical representations.
- Edge matching tools to ensure topological consistency across international boundaries.
- Quality management tools to guarantee that outputs meet user expectations and comply with defined specifications.
- Generalisation tools, both for investigating new processes (e.g., from 1:10,000 to 1:50,000, 1:100,000, and 1:250,000) and improving existing workflows for smaller scale products such as EuroGlobalMap.

These tools are essential for the new production process outlined in WP4, which integrates large scale national data into a central pan European database. WP5 ensures that data harmonisation and quality checks can be automated and scaled, reducing manual intervention and supporting sustainability.

Kadaster (NL) leads WP5 on automated generalisation and quality assurance, building on experience from national and European projects. Kadaster and IGN France both bring experience in automated generalisation, with Kadaster focusing on generalisation workflows and QA tools, and IGN France contributing expertise in edge matching and harmonisation processes. EuroGeographics provides coordination and user engagement.

WP5 is therefore a cornerstone of OME2's technical architecture, enabling the transformation of heterogeneous national datasets into harmonised, edge matched, and quality controlled pan European products.

4. Methodology and Approach

The methodology followed in this assignment was designed to evaluate the feasibility of automated generalisation for selected themes and countries using open source technologies. The work was carried out by developers from Kadaster, who jointly investigated relevant tools, designed the experimental workflows and executed the technical assessments. The approach combined iterative development, structured testing and focused innovation activities.

A dedicated innovation week, internally referred to as a High5 week, was organised to accelerate the exploratory phase of the work. During this week, the development effort was concentrated on testing various open source components, analysing their behaviour on representative datasets and identifying potential processing strategies for the generalisation tasks defined under T5.1.

The experimental workflow was developed in multiple stages. The initial phase focused on data from the Netherlands, as this dataset could be compared directly with the national reference data maintained by Kadaster. This provided a controlled environment for validating geometric transformations, testing algorithm behaviour and establishing baseline expectations for the generalisation of the themes Administrative Units, Transport and Hydrography.

After establishing the initial workflow, the scope was extended to include datasets from Belgium and France. These datasets were integrated into the same processing environment to evaluate how the workflow behaved under different data structures, levels of detail and national modelling practices. This multi country testing allowed the developers to refine and adjust the workflow, identify generalisation rules that are transferable across borders and determine where country specific adaptations were required.

Throughout the process, the work followed an iterative cycle of processing, evaluation and adjustment. The feasibility assessment was based on a combination of visual inspection, topological checks, comparison against national reference datasets where available and analysis of the completeness and stability of the resulting generalised outputs. The overall methodology aimed to provide a realistic assessment of how open source tools perform when applied to harmonised large scale data in a European context.

5. Use of Open Source Software

The generalisation activities carried out in this assignment were implemented exclusively with open source technologies. This approach ensures transparency, reproducibility and long term sustainability, and aligns with the objectives of OME2 to develop shared European processing capabilities. The workflow was built around a combination of database technology, desktop GIS software and scripting tools.

5.1. PostgreSQL

PostgreSQL is an open source relational database management system that provides robust data storage, indexing and query capabilities. It was used as the central data environment for all generalisation activities. The database enabled structured management of the large scale datasets from the Netherlands, Belgium and France and provided a stable platform for applying iterative processing steps.

5.2. PostGIS

PostGIS is an open source spatial extension for PostgreSQL that adds geometry types, spatial indexing and advanced spatial processing functions. It served as the primary engine for the generalisation tasks. PostGIS provides tools for simplification, measurement, topology checks and custom spatial transformations.

The project also made use of the PostGIS topology extension, which provides a data model for storing features with shared boundaries. This allowed the team to test generalisation procedures while preserving cross border continuity, especially for administrative units.

5.3. QGIS

QGIS is an open source desktop geographic information system used for visualisation, inspection and interactive analysis. It was used to connect directly to the PostgreSQL and PostGIS environment, display intermediate and final results and carry out visual interpretation of generalised outputs. The tool was essential for quality checks, manual validation and comparative analyses between the three countries in scope.

5.4. Python

Python is an open source programming language widely used in geospatial processing. In this assignment, Python was used for automation, orchestration of generalisation steps and batch testing. Python scripts allowed the two developers to run parameter variations, manage workflows, export intermediate results and create reproducible test sequences

5.5. SQL

SQL was used as for executing PostGIS spatial functions during the generalisation process. Rather than exporting data to external tools, all geometric and topological operations were triggered directly within the PostgreSQL environment by invoking PostGIS functions through SQL statements. This approach ensured that processing was performed close to the data and allowed full control over execution order, parameterisation and validation.

5.6. Azure flexible PostgreSQL server

The processing environment was deployed on a cloud based PostgreSQL server with flexible scaling capabilities. Although Azure itself is not open source, the hosted software stack consisted entirely of open source components. The server provided the required computational capacity for large datasets and allowed the team to scale up during intensive processing tasks.

5.7. Generalisation algorithms in open source libraries

The workflow relied on commonly available simplification algorithms implemented in open source software, including Douglas-Peucker and Visvalingam-Wyatt. These algorithms were applied to Administrative Units, Transport and Hydrography, with different parameters tested for inland boundaries, coastlines and linear transport infrastructure.

The open source approach also enabled transparent evaluation of the limitations of the available algorithms. Some operations, such as preserving topology during simplification, required the use of specific PostGIS functions or workarounds. Other operations, such as coastline processing or multi-level administrative hierarchies, required iterative tuning of the workflow.

As will be discussed in detail in the results chapters, generalising European-scale vector data using only open-source tools proved significantly more challenging than expected. The current state of open-source generalisation functionality is limited, and many operations that are available out-of-the-box in proprietary software had to be developed manually. Some of the simpler processes could be implemented successfully, but for more complex procedures the effort required exceeded the scope of this prototype and development was not pursued further.

6. Results – Administrative Units (AU)

The generalisation of Administrative Units (corresponding to ERM BND and EBM) was carried out through a sequence of data preparation steps, topology construction, simplification procedures and validation routines, all executed in PostgreSQL and PostGIS. The goal was to test whether the harmonised large scale input data from the Netherlands, Belgium and France can be transformed into topologically consistent mid-scale administrative units using only open-source tools.

The resulting tooling is a prototype SQL script that processes the input data to produce the desired generalised geometries. Due to limitations discussed earlier, no shell or user interface has been implemented to run the tool. A video demonstration has been provided, showing the script in action and explaining how to execute it.

This tooling can be executed for each of the administrative levels (1-6) and the maritime zones, or it can be run for a cumulated selection of the administrative levels as described in 6.1. It should also be noted that, in this prototype, no parameter values for generalisation have been defined for any of the map scales.

6.1. Input structure and administrative complexity

The harmonised input data reflects different national administrative hierarchies:

- France: six levels (national, regions, departments and three sub levels)
- Belgium: five levels (federal, region, province, arrondissement, municipality)
- Netherlands: three levels (national, province, municipality)

Administrative unit data exists at multiple hierarchical levels. To avoid duplication and ensure that only the highest available level per country is used, the workflow selects geometries by testing presence across descending administrative levels. A temporary schema consolidates the selected geometries.

The selection logic prioritises:

- Level 6 units when available,
- Level 5 units only for countries missing level 6,
- Level 4 similarly, and so on.

A single table of distinct country geometries is created, indexed, clustered, and vacuumed to optimise later processing.

6.2. Construction of a topological data structure

Topological validity and consistency is crucial for multi-scale generalisation. The workflow constructs an empty topology using the PostGIS topology extension, specifying the name, preferred CRS and precision:

```
SELECT CreateTopology('au_selection_topology',3035,0);
```

This creates a new database schema with tables for the faces, edges and nodes:

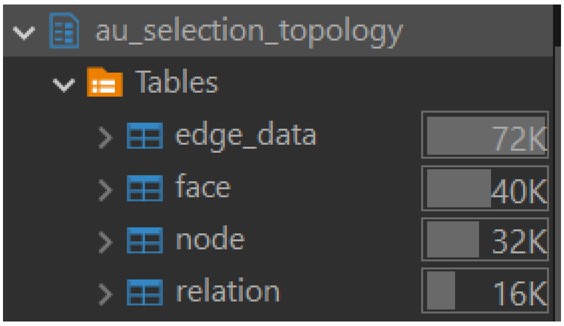

Figure 1: Database structure

To populate this topology schema, PostGIS's ST_CreateTopoGeo function is used. This loads all geometries into a topological structure with shared edges, ensuring that adjacent polygons remain consistent after simplification:

```
SELECT ST_CreateTopoGeo('au_selection_topology', st_collect(geom)) FROM
temp.au_selection;
```

A second topology is then created to store the generalised output.

6.3. Simplifying geometries

The boundaries were generalised using standard PostGIS simplification functions. Two different algorithms were applied. For inland edges, the Douglas–Peucker algorithm (ST_SimplifyPreserveTopology) was used because it emphasises preservation of major geometric features by retaining points whose removal would exceed a given deviation threshold. However, when applied to complex coastlines, Douglas–Peucker can create angular artefacts or reduce small islands to narrow slivers at higher tolerance values.

To avoid such effects on coastal geometry, the Visvalingam–Whyatt algorithm (ST_SimplifyVW) was applied instead, see Figure 2 and 3. This method simplifies geometry by iteratively removing points with the smallest effective area, resulting in a more gradual reduction in detail and typically more natural-looking coastline shapes. Although it does not explicitly preserve specific points, it tends to avoid the sharp distortions sometimes produced by Douglas–Peucker.

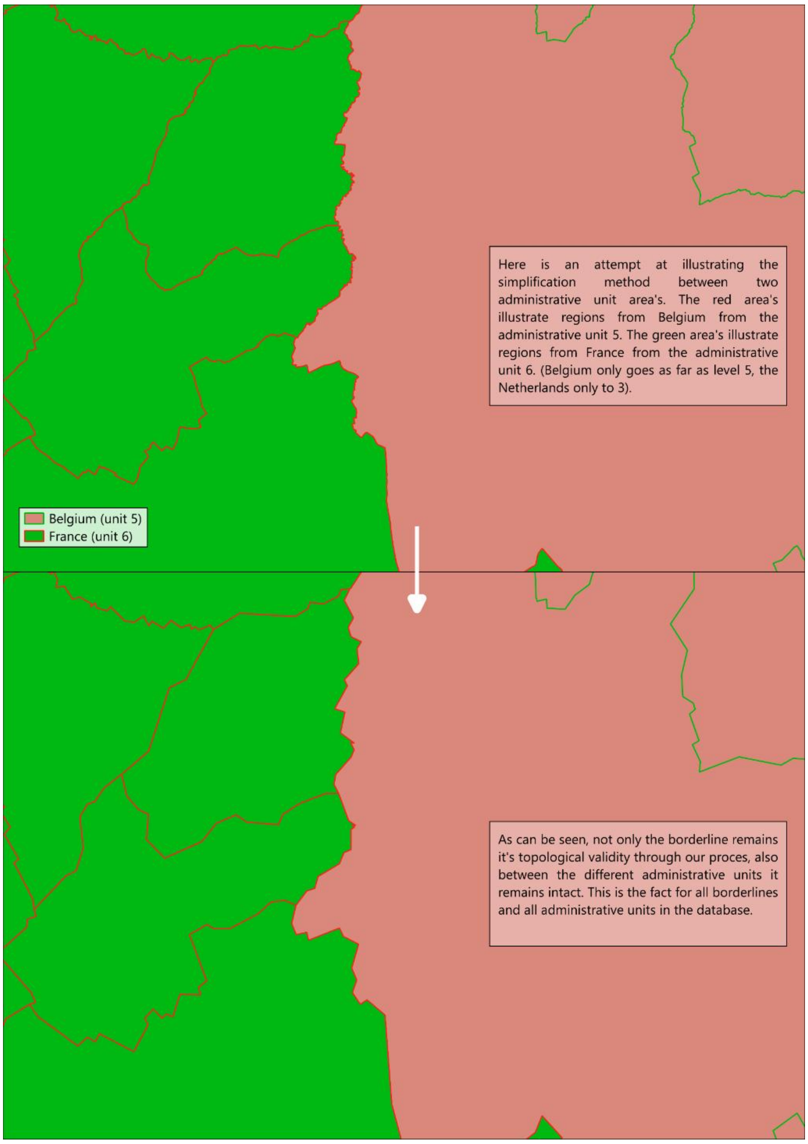

Figure 2: Results of the boundary generalisation

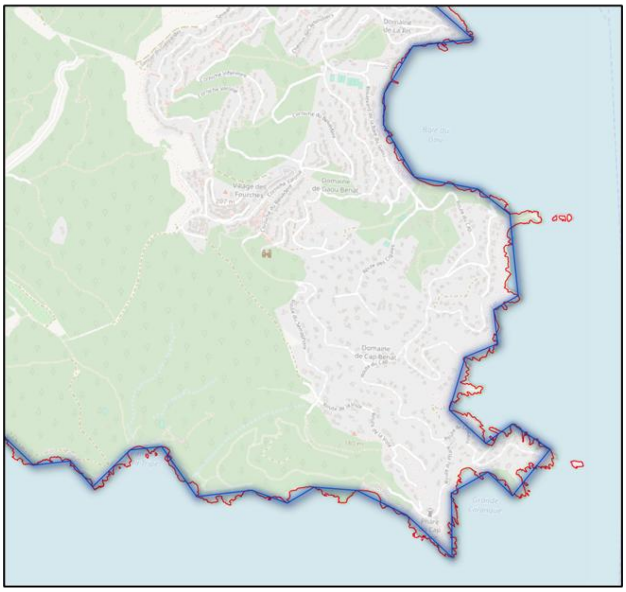

Figure 3: Results of the simplification of the coastline, note how small islands are being removed.

In PostGIS this was achieved by ‘copying’ the simplified edges in the second topology schema:

```
SELECT ST_CreateTopoGeo('au_selection_topology_simplified', geom)
FROM (
    SELECT
        ST_Collect(
            CASE
                WHEN left_face = 0 OR right_face = 0 THEN st_simplifyvw(geom, 2500)
                ELSE ST_SimplifyPreserveTopology(geom, 50)
            END
        ) AS geom
    FROM au_selection_topology.edge_data
) AS foo;
```

Figure 4: Code ST_CreateTopoGeo

6.4. Attribute Restoration

After generating simplified polygons from the topology, the geometries were re-associated with their original attribute information. This involved converting the simplified faces back into polygon geometries and performing a spatial join with the source administrative dataset. Through this join, each simplified polygon inherited the correct country and administrative attributes, ensuring consistency between the generalised layers and the original data model:

```
create table TEMP.simple_geom as (
 select st_getFaceGeometry('au_selection_topology_simplified', face_id) as geom
    from au_selection_topology_simplified.face
    where face_id > 0); -- the first row has no geometry and results in an error.
```

Figure 5: Create table TEMP.simple_geom

7. Results – TRANSPORT (TN) and HYDROGRAPHY (HY)

The generalisation of the transport theme proved significantly more complex. Several limitations of the available open-source functionality became apparent during development. As a result, only a subset of relatively straightforward operations could be implemented, while the more demanding generalisation steps could not be fully realised within this prototype.

7.1. ScaleMaster diagram

To better understand the components of the transport theme, a ScaleMaster diagram (Brewer et al., 2007) was created. A ScaleMaster diagram provides a visual overview of how each feature class in a dataset is represented and transformed across map scales. It specifies which features appear at each scale and what generalisation operations are applied, serving as a structured blueprint for scale-dependent cartography.

	50k	100k	250k
C-	Remove ‘walkway’	Remove ‘tractor road’	
	Remove ‘bicycle’	functional_road_class < 4	functional_road_class < 2
	Remove ‘pedestrian zone’		
Gc	Small roundabouts	All roundabouts	Complex junctions
Gm	Merge ‘dual carriage way’		Merge parallel ‘motorway’
Gs	Tolerance=5m	Tolerance=10m	Tolerance=30m
Go	Smoothen simplified roads	Smoothen simplified roads	Smoothen simplified roads

Figure 6: Sample of the ScaleMaster diagram for the transportation theme.

7.2. Road generalisation

The generalisation workflow for the road network is implemented as a sequence of stored procedures written in PL/pgSQL on top of PostGIS and PostGIS Topology. The process is divided into two main stages:

1. selecting an initial subset of road links, and
2. applying a series of topology-driven thinning and simplification operations within administrative partitions.

1. Initial selection of road data

The procedure `initial_selection_road_link()` creates a working copy of the road-link table and selects only the features that meet a set of predefined criteria. These include temporal validity (features must be active), specific `form_of_way` values, and a constraint on the `functional_road_class`. A temporary filter for the Netherlands is currently applied.

This step is essentially a data-reduction filter and does not generalise geometries, but it defines which roads are allowed into subsequent phases.

2. Partition-based generalisation

The core procedure, `generalize_road_link()`, iterates over administrative units (level 3). For each unit it retrieves the partition geometry and triggers the generalisation workflow only for the selected codes used for testing. This tight coupling to administrative boundaries is practical for prototyping but risks artefacts along borders if not handled carefully, and the code reflects this by checking explicitly whether edges lie on partition boundaries. Each selected partition undergoes two main operations:

a. Roundabout collapsing

`collapse_roundabouts()` identifies small, near-circular roundabouts by buffering the road segments labelled as roundabouts, constructing a minimum-bounding circle, and comparing its area to the buffer union. If a roundabout is deemed sufficiently circular and smaller than 50 m in diameter, it is collapsed into a single central connection point.

Connecting road links are extended so their endpoint snaps to the roundabout centre, after which the original roundabout arcs are removed. This avoids circular clutter at small scales, although the detection method inherently depends on buffer size and may fail on elongated or irregular roundabouts.

b. Topology-based thinning

The main reduction process is handled through the `thin_roads_by_topology()` pipeline. First, all road geometries within the partition are copied into a temporary topology (`setup_topology()`). The method extracts topological primitives (nodes, edges, faces) and stores the original attributes for later reconstruction.

A chain of operations follows:

- *Merging continuous lines*
  Nodes of degree 2 (i.e., simple continuation nodes) are detected, and their adjacent edges are merged unless they lie on a partition boundary. This reduces unnecessary segmentation but assumes that all degree-2 nodes are artefactual, which may not always be true in urban contexts.
- *Removing short dangling lines*
  Dead-end edges below a length threshold are removed, but only when they are classified as single carriageways and not located on a border. This makes the method conservative but also means it ignores certain functional classes that arguably could be removed.
- *Removing small single-edge faces*
  Faces bounded by a single edge and below a specified area threshold are eliminated. This mostly removes small loops or sliver artefacts caused by segmentation.
- *Simplifying two-edge faces*
  Faces consisting of exactly two edges are evaluated, and the shorter edge is removed if its length falls below a threshold.
- *T-junction thinning*
  Potential T-junctions (nodes of degree 3) are analysed by constructing direction vectors for each connecting edge and computing pairwise intersection angles. When one pair of edges forms a nearly straight line and the third is perpendicular, the third edge is considered a removable “stub”. If the segment is short and meets the attribute criteria, it is deleted. This is a more sophisticated operation but depends strongly on angular thresholds; results vary with road geometry quality.

Between each stage the workflow re-merges continuous lines to prevent fragmentation. See Figure 7 for an example of the results for road-thinning.

Figure 7: Example of the road-thinning.

<u>c. Reconstruction of the generalised dataset</u>

After thinning, `apply_topology_thinning()` writes the modified topology back to the working road-link table. Edges eliminated during thinning lead to the deletion of entire road-link objects. Remaining attributes are preserved by merging all topological edges belonging to each road-link object back into a single geometry.

7.3. Generalising other transportation objects

While roads form the core of the transport theme, several other feature types also require scale-dependent generalisation, including aerodromes, marker posts, ports, railways, railway-stations and runways and various "other" transport objects.

Aerodromes

The procedure for selecting aerodrome features applies a basic generalisation rule in which only the most essential objects are retained. It first filters aerodrome areas by removing features below a minimum size threshold, ensuring that only spatially significant polygons remain in the mid-scale dataset. It then selects point-based aerodromes but excludes those already represented by an area, preventing duplication across geometry types. The result is a streamlined subset of aerodrome features that preserves meaningful information while reducing unnecessary detail.

Marker posts

For linear-referencing objects such as marker posts, the workflow applies a spatial consolidation procedure aimed at reducing excessive point density while retaining representative locations. The process groups nearby marker posts into spatial clusters using a density-based algorithm (DBSCAN) with a configurable distance threshold, ensuring that only posts within meaningful proximity are aggregated. For each resulting cluster, a centroid is computed and the marker post closest to this centroid is selected as the most representative instance of that group. The output is therefore a filtered set of marker posts in which local redundancies are removed, while the overall spatial distribution of the original dataset is preserved (see Figure 8).

Figure 8: Grouping of marker posts.

Ports

The port-selection procedure streamlines the dataset by separating ports represented as areas from those represented as points. First, port areas larger than a minimum size threshold are retained, ensuring that only spatially meaningful polygons remain. Point features are then added only when they do not fall within one of the selected port areas, preventing duplication between the two geometry types. This produces a consistent, non-redundant set of ports suitable for subsequent generalisation.

Railway stations

The procedure for railway stations extracts point features located within proximity of the railway network. For each administrative partition, a buffer is generated around relevant railway links, and only stations falling within these buffered areas are retained. This ensures that the dataset includes stations functionally connected to the railway infrastructure, while minimizing extraneous or isolated features.

Road Service

The processing of road service areas focuses on reducing detail while preserving essential spatial structure. Initially, all relevant areas are selected into a working dataset. Nearby or overlapping areas are then merged to create larger, continuous features. Small or narrow areas are removed to eliminate insignificant details, and the remaining geometries are simplified and smoothed. This approach produces cleaner, more uniform representations suitable for generalized mapping without retaining excessive detail.

7.4. Conclusions

Using only open-source tools to generalise vector data at a European scale proved far more complex than anticipated. Existing open-source solutions offer limited functionality, meaning many operations commonly available in proprietary software had to be built from scratch. While some straightforward processes were successfully implemented, the effort required for more advanced procedures exceeded the scope of this prototype, and their development was not continued. Consequently, no dedicated tooling was created for hydrography; however, the tools developed for transportation could be adapted for use with that theme.

8. AI-Based Generalisation

8.1. Context

Although investigating AI-based generalisation was not an official task for OME2, Kadaster decided to review the latest scientific literature to understand what is currently possible with AI techniques. This was a desk study based on published research papers, not practical testing. The goal was to identify capabilities, limitations, and potential directions for future projects—without changing OME2's focus on authoritative, harmonised datasets.

OME2 prioritises quality, interoperability, and compliance with specifications, using proven rule-based tools for harmonisation and edge-matching. AI-based generalisation is discussed as a way to automate scale reduction and improve efficiency, but its integration must be carefully evaluated against these requirements.

8.2. What We Can Do

AI methods such as deep learning and generative models have demonstrated promising results in research and pilot projects. For example, neural networks can simplify building footprints for smaller scales while preserving their overall shape. Generative adversarial networks (GANs) are capable of producing visually appealing linework for mountain roads and rivers, reducing clutter and improving legibility. Additionally, AI can apply style transfer to create map-like images from aerial photographs, which is particularly useful for rapid prototyping and visualisation in previews or communication contexts.

8.3. What We Cannot Do

AI alone cannot yet produce authoritative datasets because OME2 requires strict compliance with specifications, topology integrity, and harmonisation across borders. These standards cannot be guaranteed by AI without extensive postprocessing. Likewise, rule-based harmonisation and edge matching remain essential for interoperability and legal compliance since AI is not yet capable of reliably handling complex semantic and topological constraints. Finally, fully automating cross border integration or cadastral data processing is not feasible because these tasks depend on authoritative sources and legal frameworks rather than AI inference.

8.4. Risks and Challenges

AI-generated geometries can lead to topology breakage, introducing disconnections or overlaps. Deep learning models often function as opaque systems, making it difficult to trace or reproduce decisions. The performance of AI strongly depends on the quality and diversity of training data, which means rare patterns may be misrepresented. Additionally, ethical concerns arise because AI-generated maps can mislead users if their origin is not clearly disclosed.

8.5. Future Direction

AI is not part of OME2's core deliverables, but it offers opportunities for future enhancements:

- Hybrid workflow: AI proposes generalisation; rule-based tools validate and enforce constraints; humans approve.
- Pattern recognition: AI can help identify patterns to optimise conversion parameters for different geographic contexts.

Promising research areas include vector-native deep learning for geometry fidelity and GANs with topology-aware constraints for visual generalisation.

8.6. Key Message

AI-based generalisation offers interesting possibilities, but for OME2 the priority remains authoritative, harmonised datasets using proven rule-based methods. AI exploration should be considered as a future enhancement, not as part of the current project scope.

9. Final conclusions

Open-Source Feasibility

The work carried out within OME2 shows that relying exclusively on open-source technology for automated generalisation at a European scale is currently not sufficient. Some processes were successfully implemented using PostgreSQL/PostGIS, QGIS, SQL, and Python. However, advanced functionality and integrated workflows are missing, making a fully operational, scalable solution unrealistic at this stage. Open source could become a viable option in the future, but it would require multiple years of development and investment to reach the level of maturity needed for production environments.

AI-Based Generalisation

Artificial Intelligence (AI) offers promising opportunities for the future. Techniques such as deep learning and generative models can simplify geometries and improve cartographic representation. AI cannot yet deliver authoritative, topologically consistent datasets that meet European harmonisation and quality standards. Future projects should explore hybrid workflows combining AI proposals with rule-based quality assurance and human validation.

Role of ESRI Software

For operational production, ESRI software has historically provided a proven and effective solution, with its generalisation capabilities co-developed together with Kadaster approximately ten years ago.

This collaboration resulted in a fully automated workflow that significantly reduced production time and set a high standard for efficiency at the national level. Unlike open source, ESRI offers ready-to-use tooling today, which can be leveraged immediately for operational needs. While OME2 focuses on open-source and interoperability for long-term sustainability, this experience highlights the importance mature tooling in achieving operational success.