# Logic and Methods of the SDMX Semantic Code List Generator

## General Workflow

The **SDMX\_CL\_Generator** project implements the full processing cycle of SDMX code lists: from loading source files to generating RDF models in TTL format. The pipeline includes the following stages:

* **Source loading**: Code list sources are described in a CSV table (*Code\_Lists\_GR\_SDMX.csv*). For each list, the agency, identifier, and XML link are specified. An additional module ensures the transformation of URNs into valid SDMX URLs. This approach enables centralized management of source references and easy extensibility.
* **Parsing and data consolidation**: SDMX 3.0 XML files are processed using the *parse\_codelist\_v3* function. The output consists of tables:
  + with codes and their descriptions,
  + with code list characteristics (identifier, agency, version, number of codes, etc.).

The strictly defined table structure ensures compatibility and robustness of subsequent analysis procedures.

* **Analysis and grouping**: The analytics module generates reports on code uniqueness and overlaps, classifies code lists based on predefined templates, and saves results in aggregated tables (*all\_cl\_data.csv, cl\_table.csv*). This helps identify duplicate and problematic code lists before RDF generation.
* **RDF/TTL generation**: For each code list, a SKOS scheme and a set of concepts are created. Stable URIs based on PURL are used as identifiers. All prefixes and namespaces are centrally defined in templates, ensuring consistency and reproducibility. Final data is published both as a consolidated *code.ttl* file and as separate TTL files for each scheme.
* **Quality control**: The final stage includes automated quality checks of RDF graphs. The structure, presence of mandatory attributes (*prefLabel, notation, inScheme*), usage of prefixes, and related entities are evaluated. Results are expressed as numerical quality and value indicators, enabling objective verification of compliance with set requirements.

## Rationale of the Applied Methods

* Parsing SDMX with **ElementTree** ensures independence from external libraries and guarantees high performance when processing large XML datasets.
* Using analytical metrics helps detect overlapping and duplicate code lists, reducing the likelihood of errors in final models.
* A template-based grouping approach makes classification manageable and reproducible, eliminating subjective decisions.
* Centralized URI and prefix management guarantees uniformity and long-term stability of published data.
* Automated quality control reduces manual verification costs and provides transparent acceptance criteria for publication.

## Efficiency and Effectiveness

* The project successfully ensures the full data transformation cycle: from raw SDMX XML to validated RDF models.
* The applied methods enable rapid scaling to new agencies and code lists without changing the core logic.
* The quality control system generates objective indicators, making the publication process both manageable and measurable.
* The resulting RDF models comply with **FAIR** principles and **Linked Open Data**, confirming the practical value of the solutions.