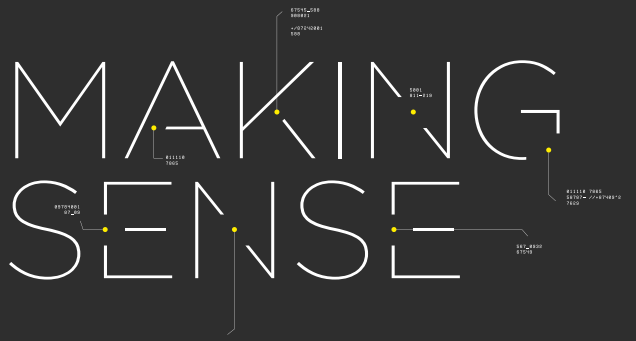
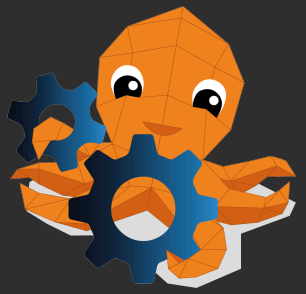
A practical SDMX /VTL implementation using Trevas and FMR

Glenn Tice, BIS Monetary and Economics IT / Nicolas Laval, Making Sense

2024 SDMX Experts Workshop



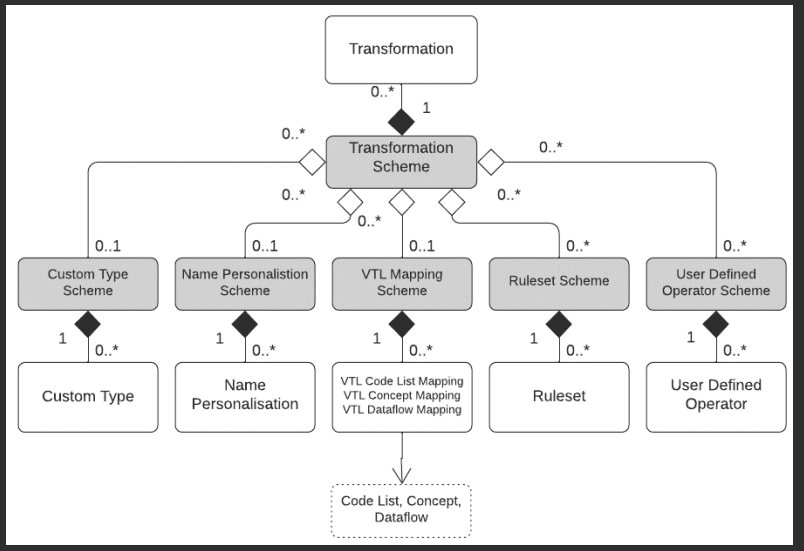


VTL INTEGRATION WITH SDMX

- VTL has its own information model based on GSIM, but is designed to be compatible with SDMX and other standards like DDI
- The SDMX Section 6 Technical Notes explains how VTL and SDMX work together
- Two important elements for a practical implementation are:
    - How VTL programs can be stored and exchanged together with the SDMX structures of the datasets involved
    - How SDMX artefacts can be referenced in VTL code as inputs and / or outputs of Transformations and for validation Rules

STORAGE AND EXCHANGE OF VTL

- In the SDMX context, VTL code is managed as structural metadata
- The SDMX information model provides several artefacts for this purpose
- Centres on the **Transformation Scheme** artefact which **encapsulates a VTL program**
- Allows VTL to be stored and exchanged in the same way as other SDMX structures



REFERENCING SDMX ARTEFACTS IN VTL STATEMENTS AND RULES

- SDMX artefacts can be referenced in VTL code using full or partial URNs

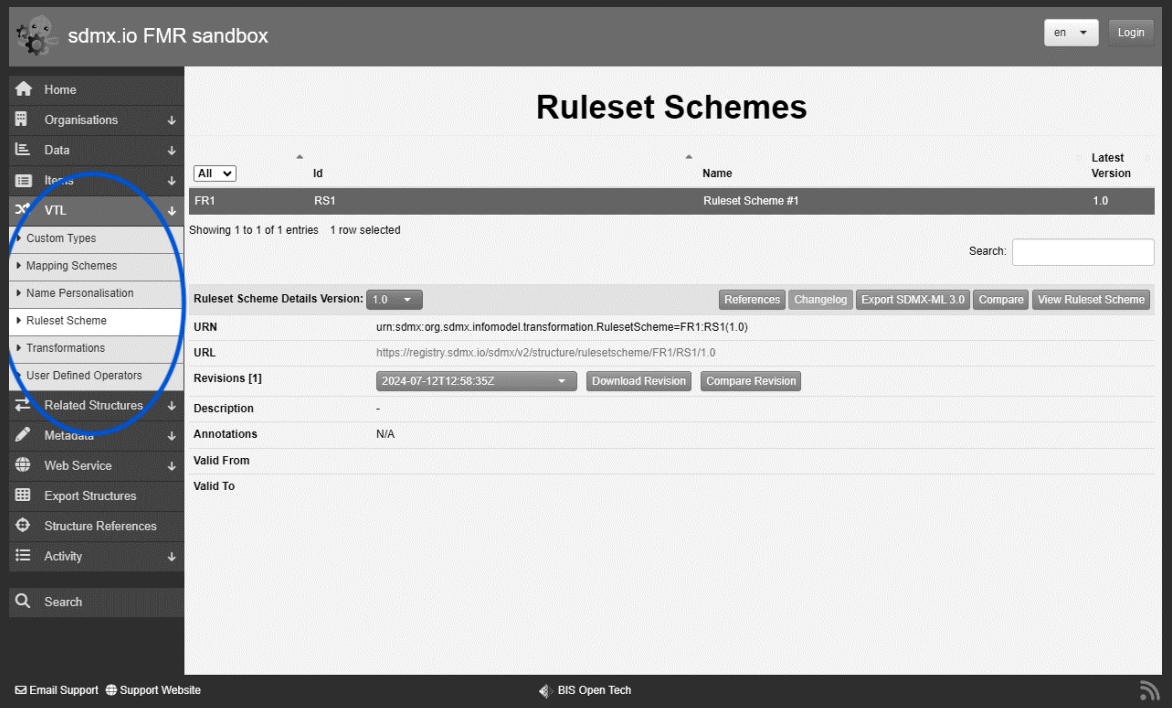
```
DS_r := "urn:sdmx:org.sdmx.infomodel.datastructure.Dataflow=ECB:EXR(1.0)" / 10;

define datapoint ruleset DPR_1 (valuedomain CL_AREA as A, CL_CURRENCY as B) is
    when expression then "..."
end datapoint ruleset;
```

- Or, by user defined aliases defined using VTL Mapping Schemes which allow SDMX artefacts to be given symbolic names

```
DS_r := EXCHANGE_RATES / 10;
```

FMR SUPPORT FOR STORAGE AND EXCHANGE OF VTL



- All VTL structures supported in FMR 11
- VTL syntax checking on structure submission
- Basic authoring and maintenance using the FMR web user interface
- Better VTL code development environments can interface to the FMR repo by the REST API

FMR SUPPORT FOR REFERENCING SDMX ARTEFACTS IN VTL CODE

DEEP STRUCTURE CROSS REFERENCING

- FMR identifies SDMX artefacts referenced by full or partial URN in the VTL code
- Manages them as structural metadata cross references

WHY?

- Execution engines can efficiently retrieve both the VTL program and structural metadata for input datasets and validation rules in a single REST web service call for the Transformation Scheme with references=descendants
- Enforces metadata referential integrity

FMR SUPPORT FOR REFERENCING SDMX ARTEFACTS IN VTL CODE - EXAMPLE

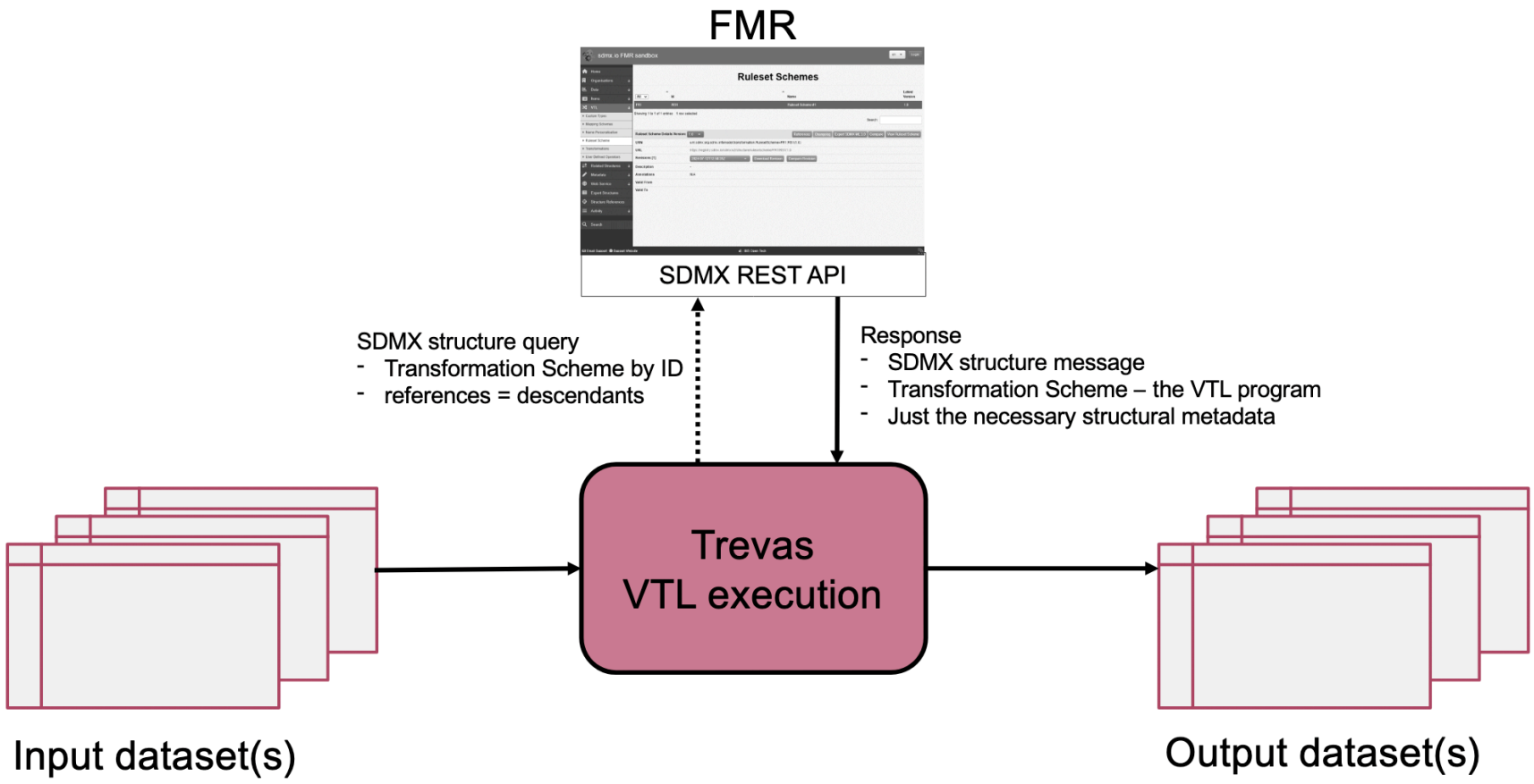
- A Transformation Scheme is constructed with a single Transformation

```
DS_r <- "ECB:EXR(1.0)" / 10;
```

On submission of the Transformation Scheme to the metadata registry, FMR will:

- Syntactically analyse the VTL code
- Identify ECB:EXR(1.0) as an SDMX partial URN
- Infer that it is a Dataflow from the code context
- Verify that the Dataflow exists in the registry
- Create a cross reference to the Dataflow from the Transformation Scheme
- Store the Transformation Scheme

EXECUTING A VTL PROGRAM USING TREVAS AND FMR



https://registry.my.org/sdmx/v2/structure/transformationscheme/all/MY_VTL_TRANS/+/?references=descendants

TREVAS

VTL ENGINE

- Java VTL engine
- Distributed computing (Apache Spark)
- Open source
- Modular

TREVAS OPEN SOURCE GALAXY

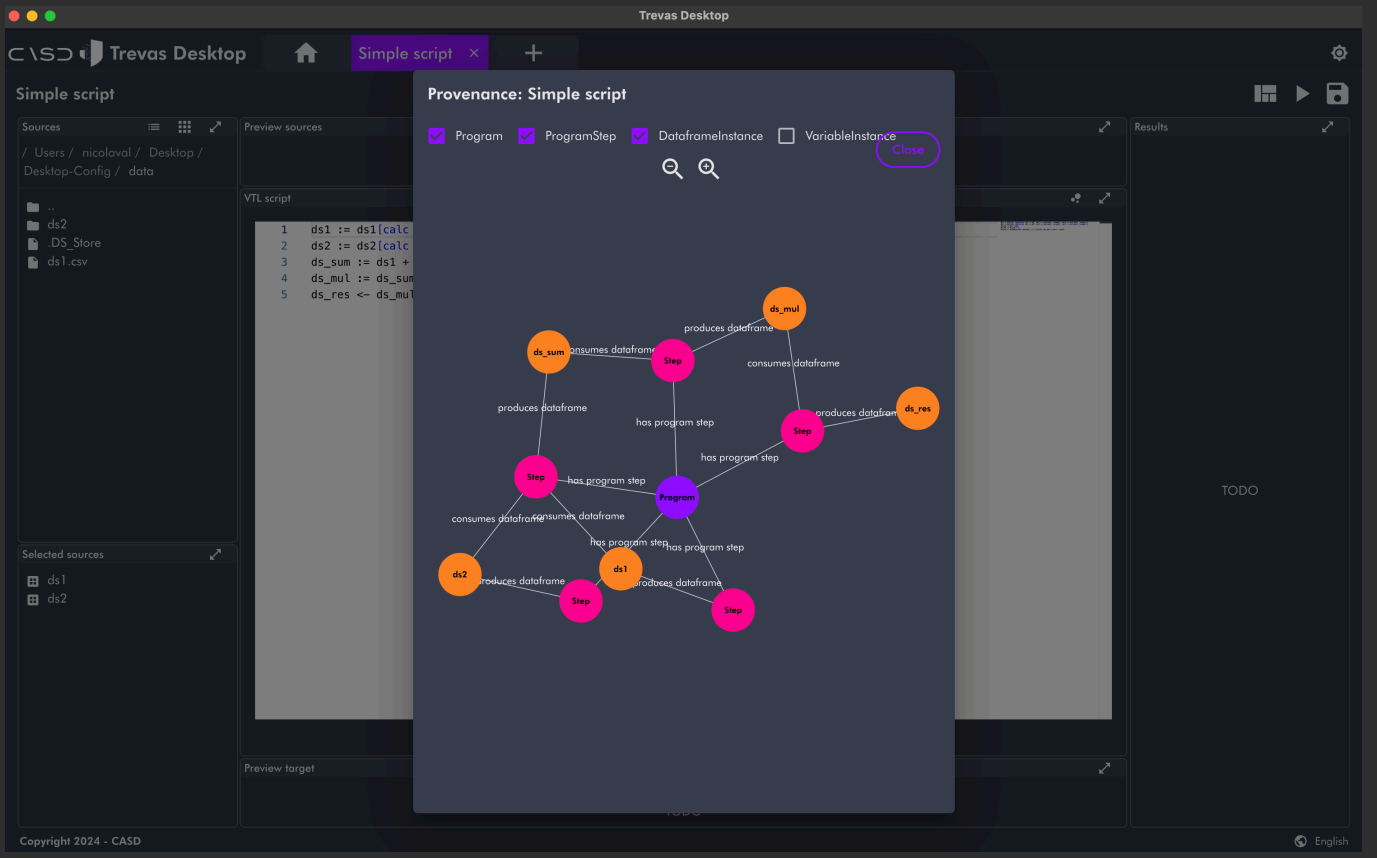
- Trevas ecosystem
- Trevas TS

- VTL editor

BUILDING SOLUTIONS WITH TREVAS

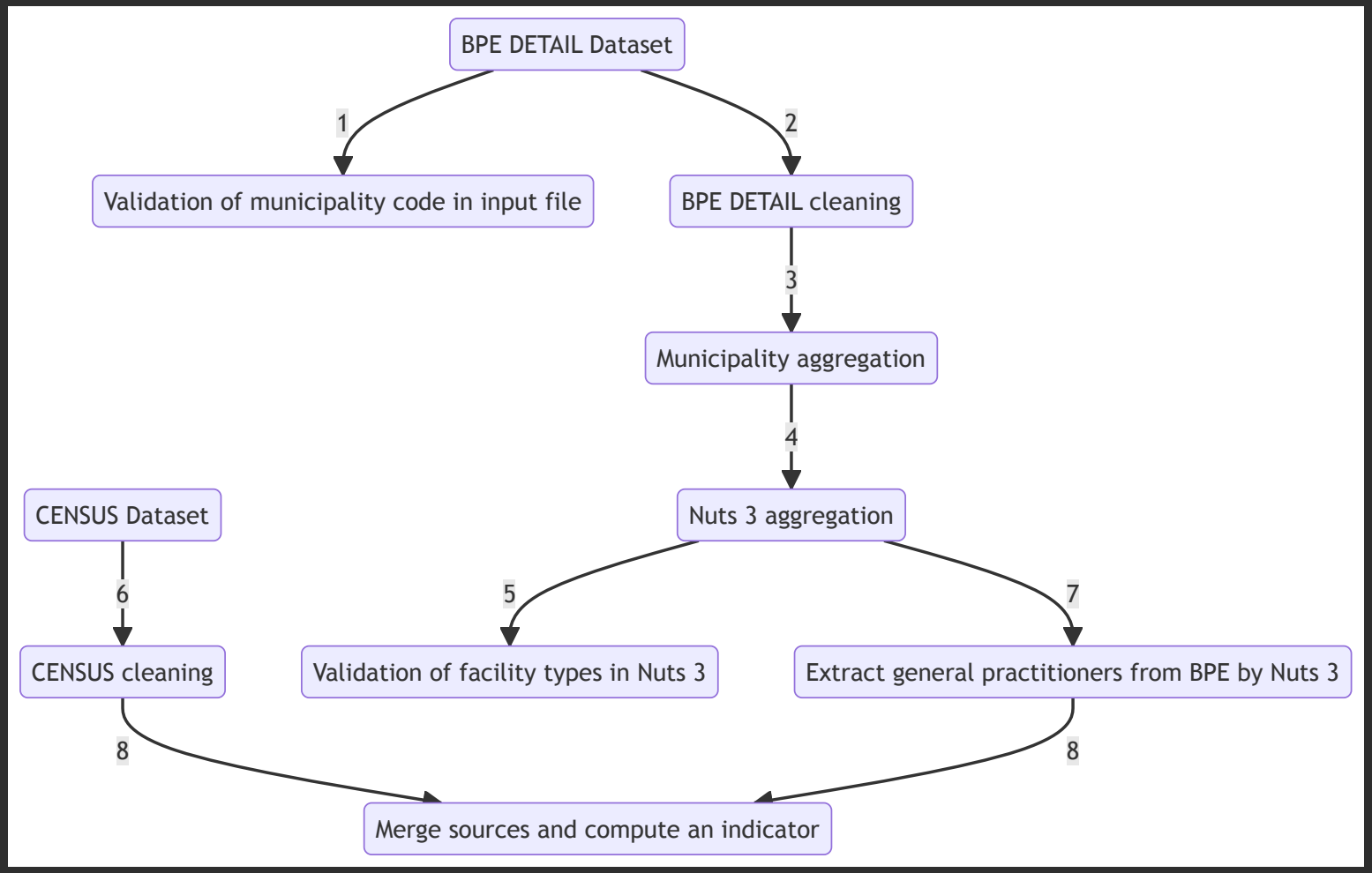
- Desktop app
- Validating data using DDI

Example:



INSEE VTL/SDMX USE CASE

Permanent Database of Facilities


BPE DETAIL Dataset
1
2
Validation of municipality code in input file
BPE DETAIL cleaning
3
Municipality aggregation
4
CENSUS Dataset
Nuts 3 aggregation
6
5
7
CENSUS cleaning
Validation of facility types in Nuts 3
Extract general practitioners from BPE by Nuts 3
8
8
Merge sources and compute an indicator

- (1) validation of municipality code in input file
- (2) clean BPE input database
- (3) BPE aggregation by municipality, type and year
- (4) BPE aggregation by NUTS 3, type and year
- (5) BPE validation of facility types by NUTS 3
- (6) prepare 2021 census dataset by NUTS 3
- (7) extract dataset on general practitioners from BPE by NUTS 3 in 2021
- (8) merge practitioners & legal population datasets by NUTS 3 in 2021 and compute an indicator

USE CASE IN ACTION

Trevas Jupyter

Onboarding sample

NEXT STEPS

OPEN COLLABORATION,

BENEFICIAL FOR ALL

- BIS: improve FMR
- Making Sense: improve Trevas SDMX support
- UNSD: POC VTL integration

Thanks, any questions?