Hersonissos, 27th May 2024

5th International Workshop On Knowledge Graph Construction
co-located with the ESWC 2024

Not Everybody Speaks RDF: Knowledge Conversion between Different Data Representations

Mario Scrocca, Alessio Carenini, Marco Grassi, Marco Comerio, Irene Celino
mario.scrocca@cefriel.com

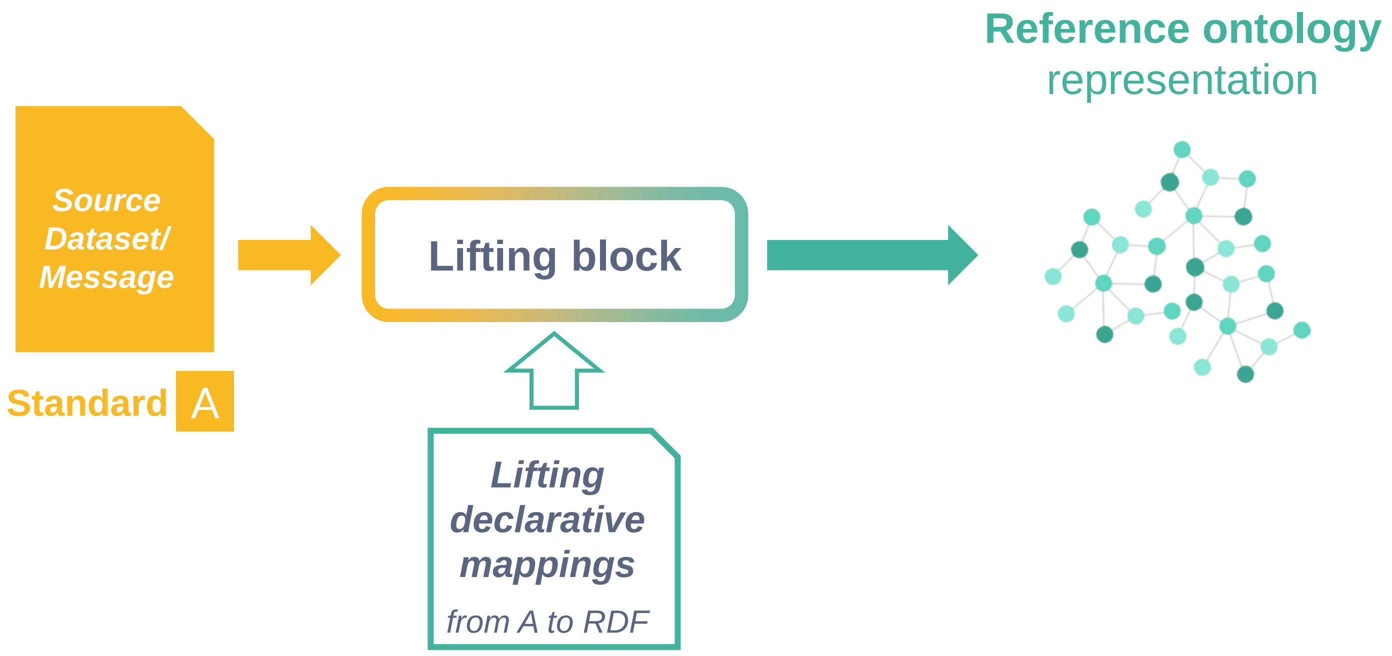
INNOVATE › GROW › REPEAT

Cefriel
POLITECNICO DI MILANO

KG Construction with Declarative Mappings

Declarative mappings emerged as a reliable, reproducible and maintainable solution for **Knowledge Graph Construction** [1].

Lifting mappings «extract knowledge» from the input generating an RDF output according to a reference ontology.

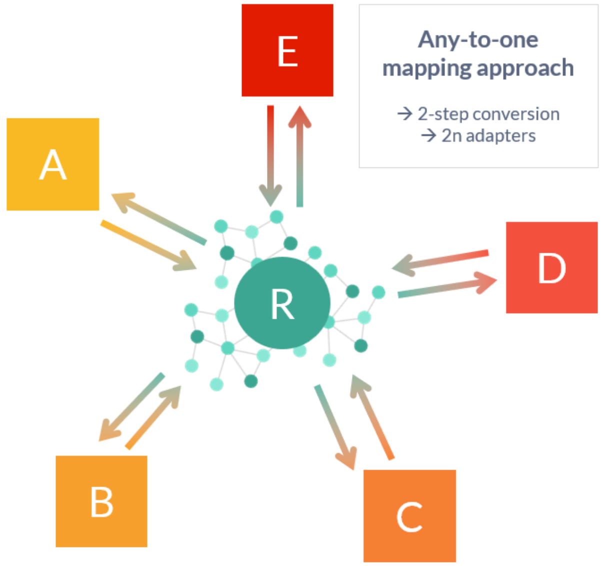


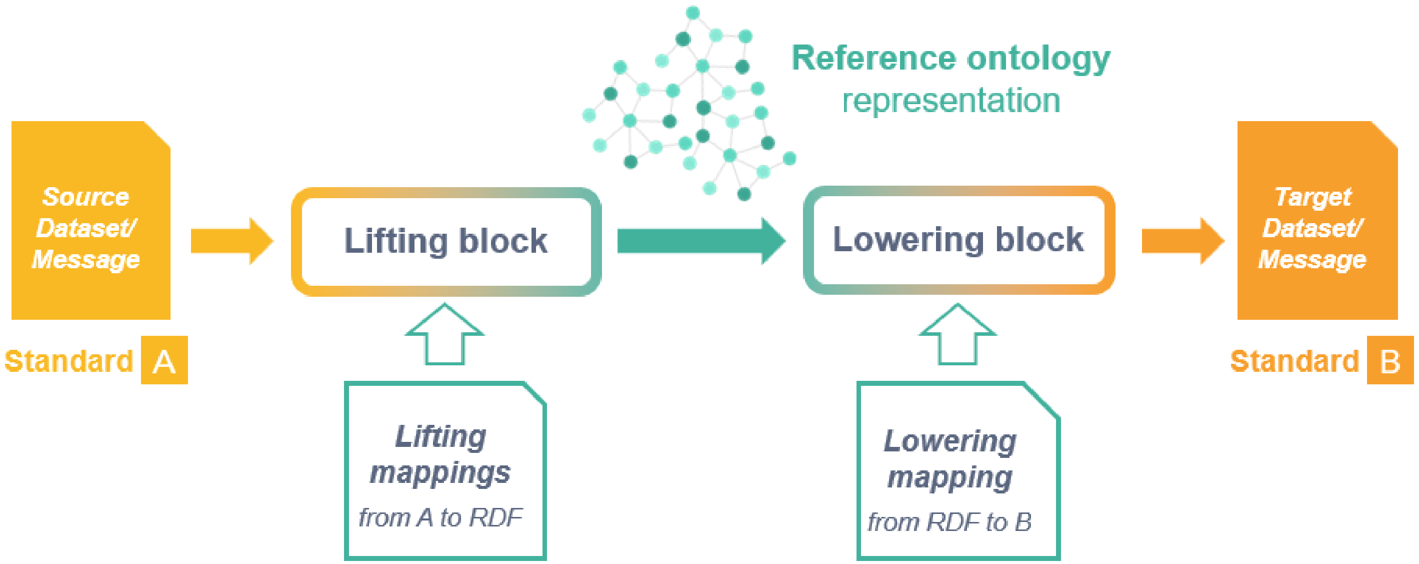
[1] D. Van Assche, T. Delva, G. Haesendonck, P. Heyvaert, B. De Meester, A. Dimou, **Declarative RDF graph generation from heterogeneous (semi-)structured data: A systematic literature review**, Web Semant. 75 (2023). doi:10.1016/j.websem.2022.100753.

But… not everybody speaks RDF!

The RDF representation enable **interoperability and data fusion** among different stakeholders. However, the **target systems may not be able to "speak" RDF**.

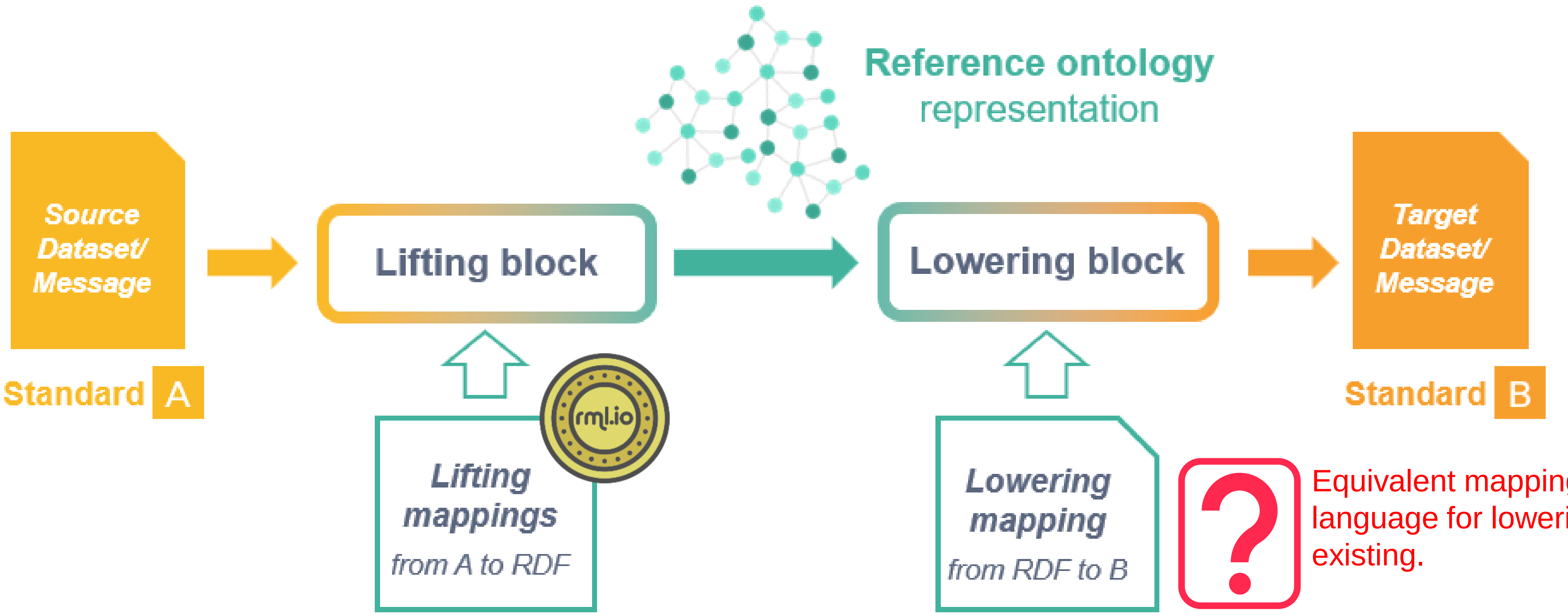
Lowering mappings define how to «access knowledge» to build the output message in the target standard. We proposed a solution to execute **semantic conversion** rules among heterogeneous information systems [2].





[2] Scrocca M., Comerio M., Carenini A., Celino I. (2020) **Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph**. In: The Semantic Web – ISWC 2020. Springer. https://doi.org/10.1007/978-3-030-62466-8_26

A lowering solution?

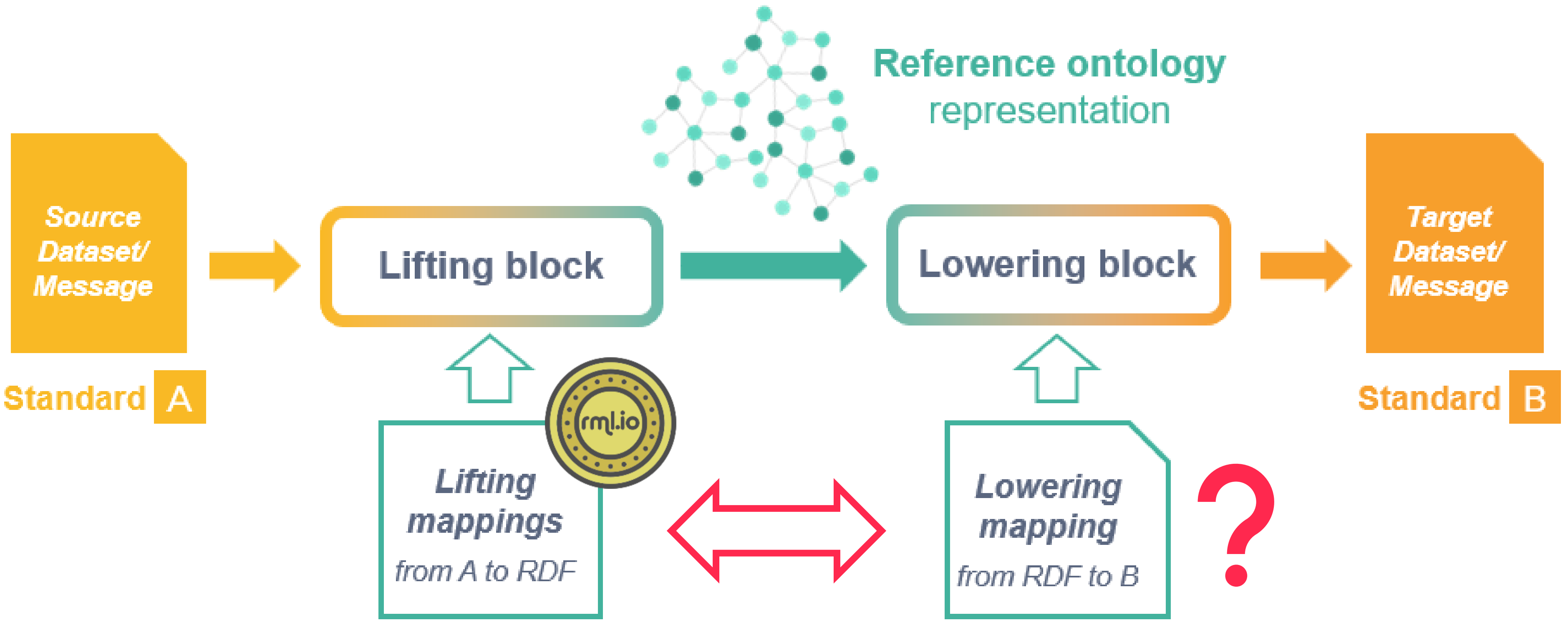


Equivalent mapping language for lowering not existing.

Only tools from RDF to specific target output (e.g., XML).

[2] Scrocca M., Comerio M., Carenini A., Celino I. (2020) **Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph**. In: The Semantic Web – ISWC 2020. Springer. https://doi.org/10.1007/978-3-030-62466-8_26

A lowering solution?



"**Reverse RML**" engine to execute the same mappings for lifting and lowering?
Did not turn out well… but that's another story!

[2] Scrocca M., Comerio M., Carenini A., Celino I. (2020) **Turning Transport Data to Comply with EU Standards While Enabling a Multimodal Transport Knowledge Graph**. In: The Semantic Web – ISWC 2020. Springer. https://doi.org/10.1007/978-3-030-62466-8_26

The initial solution…
Velocity+SPARQL Lowering

- An approach based on **templates** to guarantee flexibility on the output format
- It exploits **Apache Velocity** templates (https://velocity.apache.org) replacing at runtime variables with actual values
- **SPARQL queries** allows defining in the template how to **access an RDF Graph**
- **Velocity Template Language** (VTL) allows defining in the template how to manipulate results obtained from queries and fill the template to **generate the expected output data format**

```
#set ( $query = "
    SELECT ?id ?name
    WHERE {
        ?a <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
            <https://w3id.org/transmodel/terms#Authority> .
        ?a <https://w3id.org/transmodel/terms#name> ?name .
        ?a <https://w3id.org/transmodel/terms#id> ?id . })
#set ( $authorities = $reader.executeQuery($query) )
<?xml version="1.0" encoding="iso-8859-1"?>
<PublicationDelivery version="1.0"
xsi:schemaLocation="http://www.netex.org.uk/netex../../../xsd/
xmlns="http://www.netex.org.uk/netex" xmins:xsi="http://www.w3
    <dataObjects>
      <ResourceFrame>
        <organisations>
            #foreach($authority in $authorities)
            <Authority id="$authority.id">
              <Name>$authority.name</Name>
            </Authority>
            #end
        </organisations>
      </ResourceFrame>
    </dataObjects>
</PublicationDelivery>
```

Mappings via template-based solution

PROs

- ✓ **Decoupling of the mapping rules** (template file) **from the execution engine** (template engine)
- ✓ Flexibility towards **any textual-based output** leveraging the template language
- ✓ **Good performance and scalability** of the conversion process due to the template engine optimisation and possibility to introduce custom optimizations
- ✓ Given a set of SPARQL queries to extract the required information, **no prior-RDF knowledge needed by users to define the mapping rules**

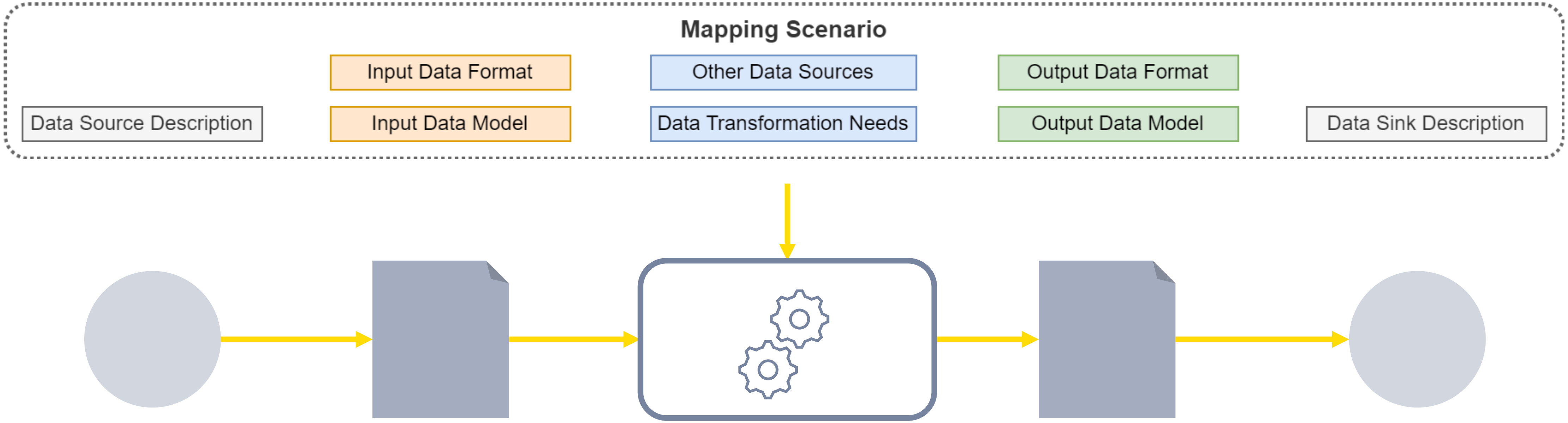
CONs

- Supports **only lowering from RDF**
- **Not "well-defined" declarative language** to express the mapping rules

Leverage state-of-the-art on declarative KG construction to define:

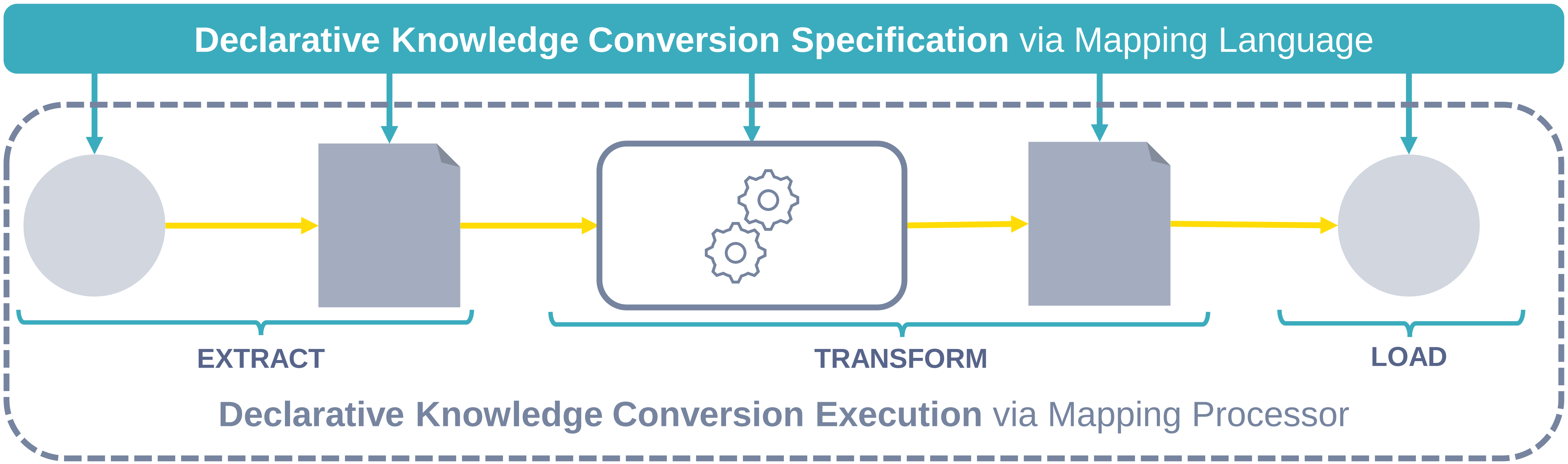
1. a **workflow** for **knowledge conversion between different data representations**
2. a **template-based tool** implementing the workflow

Mapping Scenario



Declarative Knowledge Conversion

Given a **Mapping Scenario**, we want to define a **declarative knowledge conversion** process between different data representations enabled by a **Mapping Language** and a **Mapping Processor** supporting it.



Data Frame Abstraction

R2RML

SQLTable

Table accessed by Column/Template

SQLQuery

SQL Query

Result Set (**Table**)

RML

Tabular (e.g., CSV)

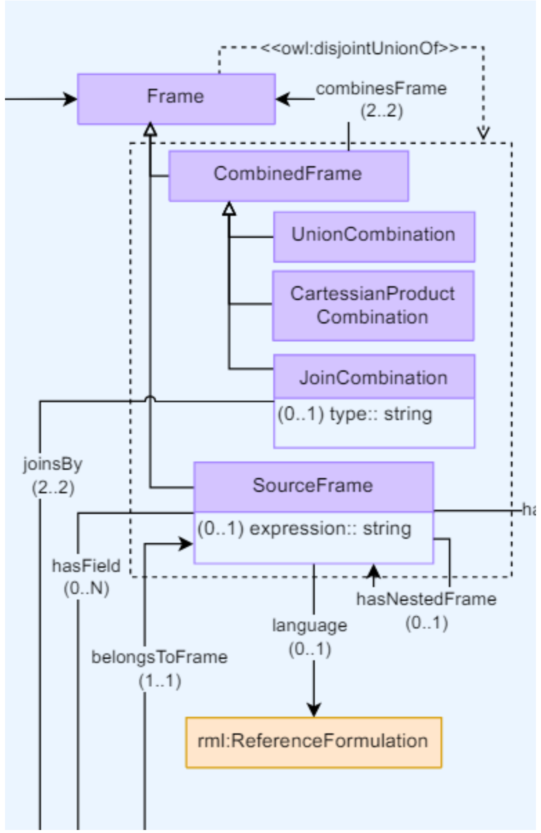
Table accessed by Reference/Template

Nested data sources (e.g., XML/JSON)

Iterator (+ RML Fields)

Table: one row for each result of the iterator
Column: access subfields

Conceptual Mapping Ontology [3]



[3] A. Iglesias-Molina, A. Cimmino, E. Ruckhaus, D. Chaves-Fraga, R. García-Castro, O. Corcho, **An ontological approach for representing declarative mapping languages**, Semantic Web 15 (2024) 191–221. doi:10.3233/SW-223224.

Data Frame Abstraction

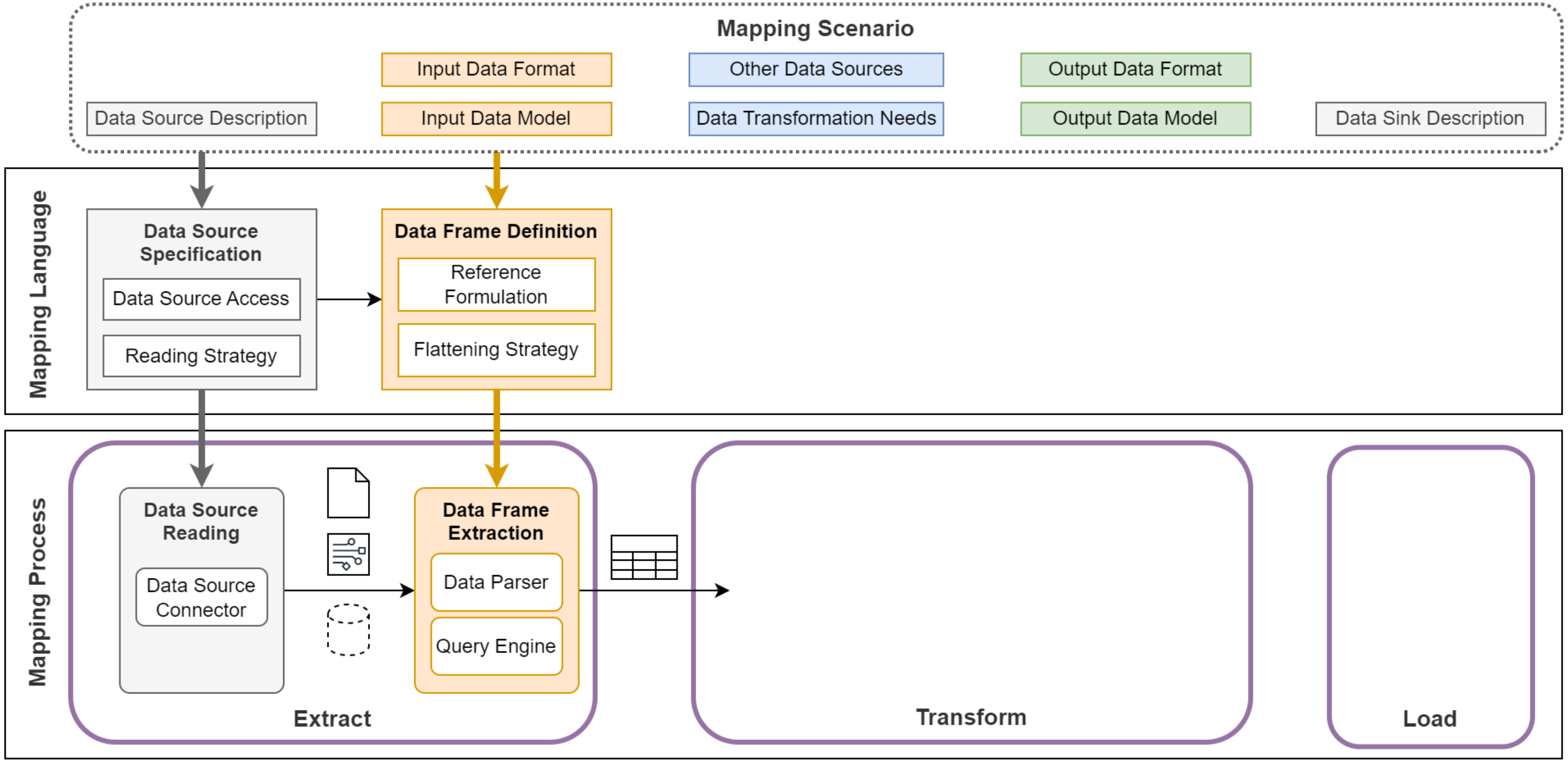
Design decision:

- Explicitly define a **data frame as intermediate abstraction** on which the declarative schema transformation rules are defined
- Assume a **fully flattened data frame**, i.e., nested data structures are mapped to a data frame in which each row already contains the values to be used during the mapping rules execution
- The **data transformation rules** are defined and applied on the data frame
- A **"combined" data frame** should be declared if the mapping rules target data from different data frames

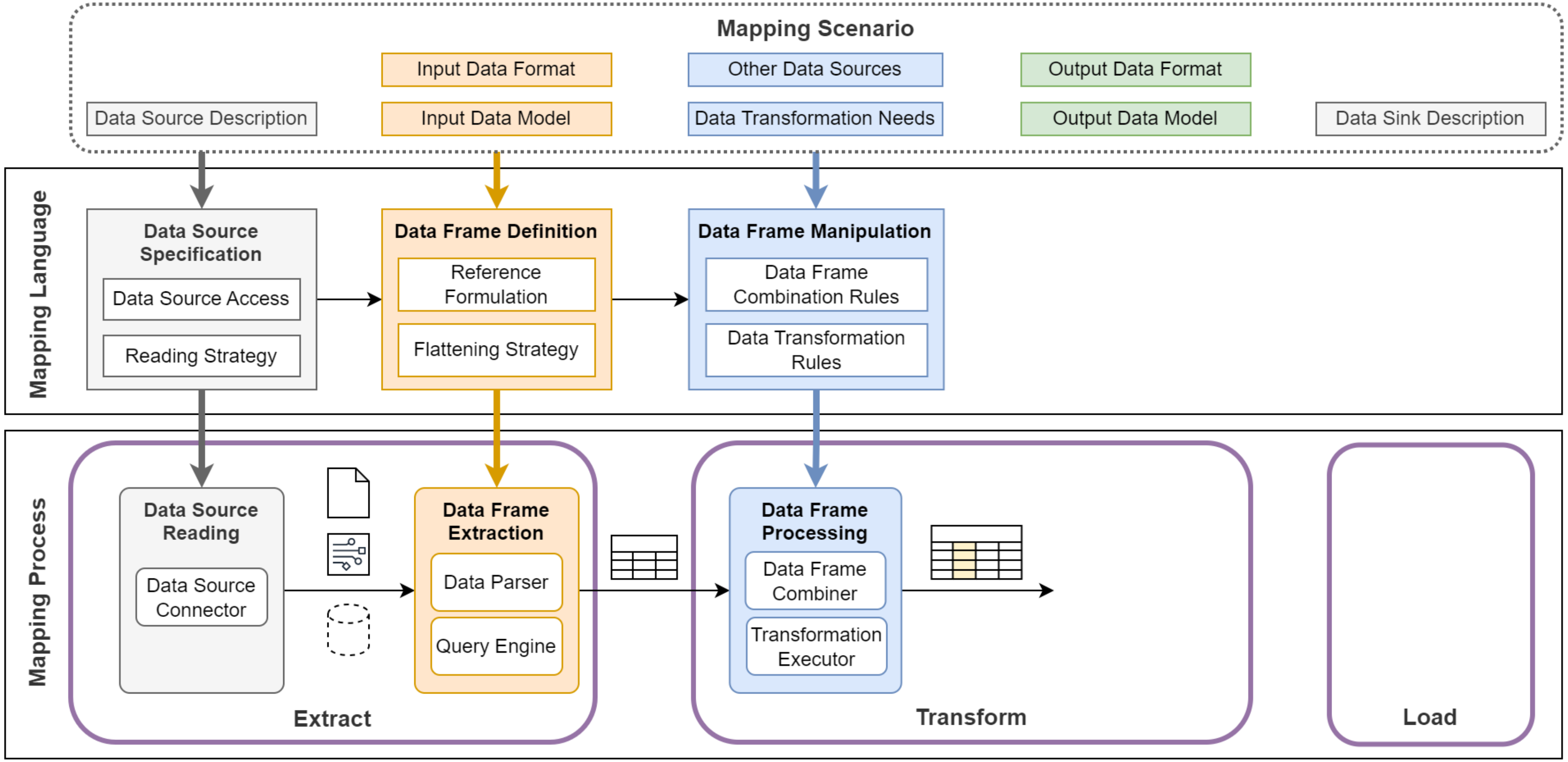
Advantages:

- Enable **better decoupling** and potential **optimizations in the execution** of mapping rules (e.g., data access)

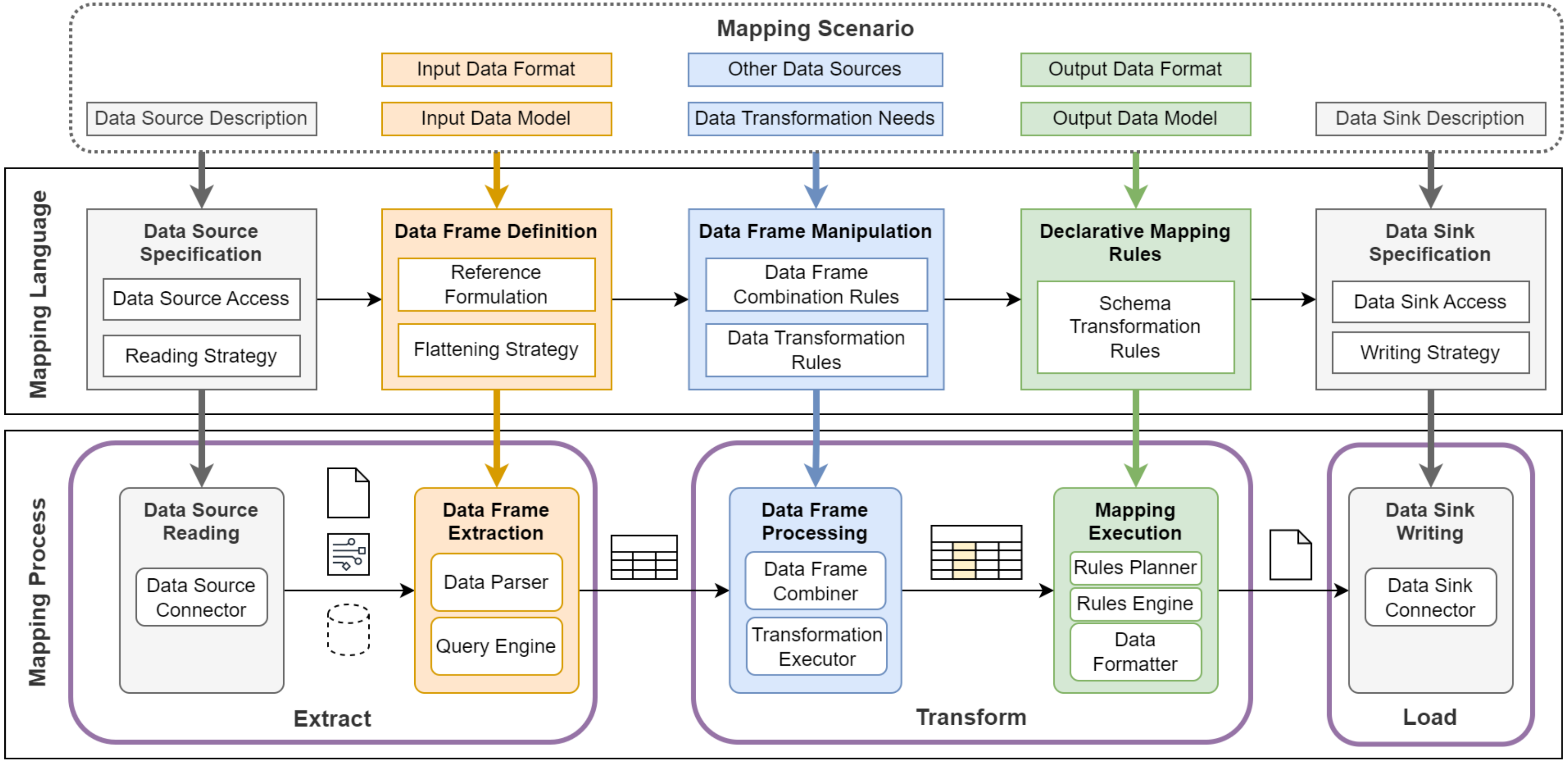
Mapping Workflow



Mapping Workflow



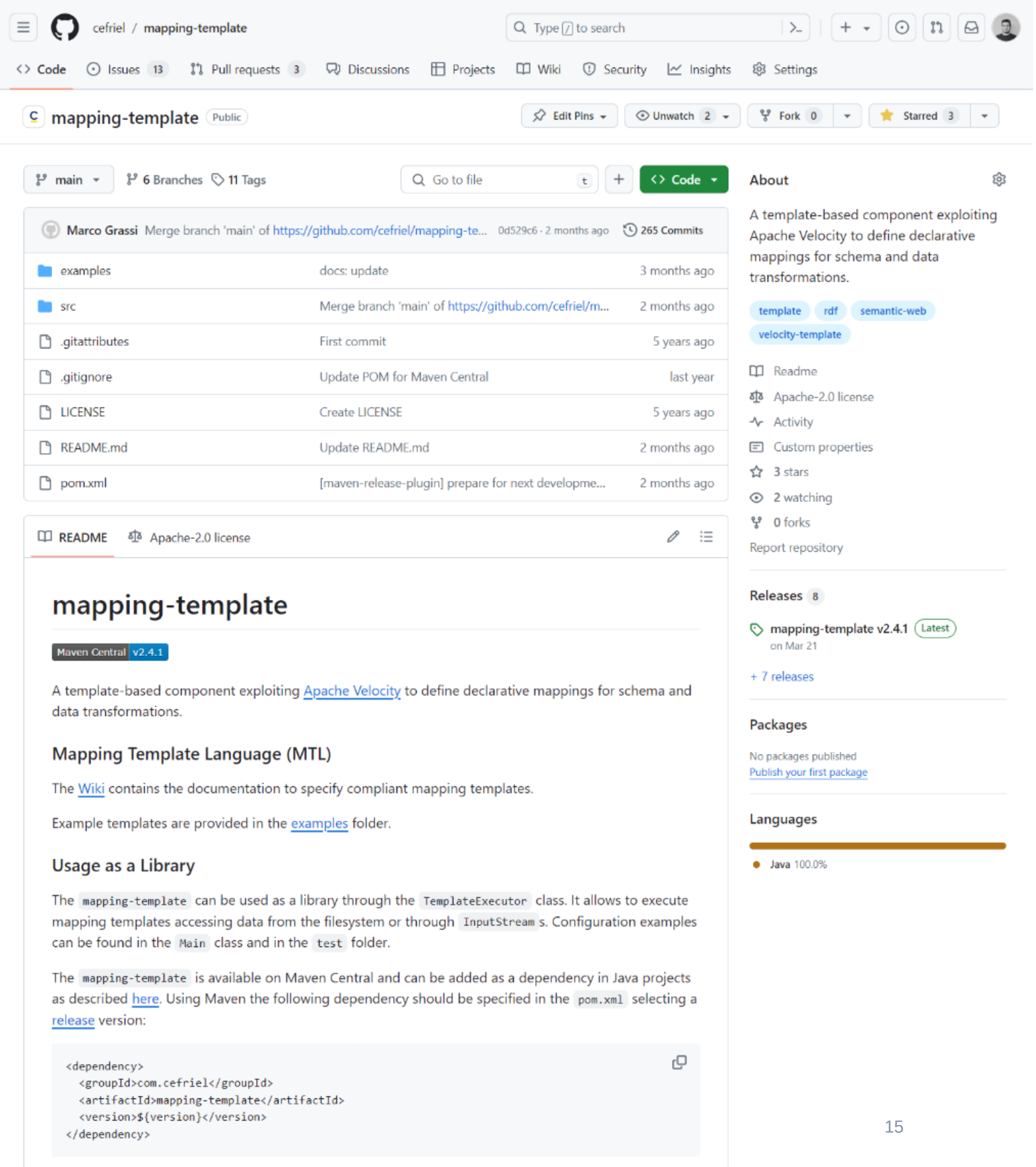
Mapping Workflow



mapping-template

Open-source software tool based on the proposed workflow and the Apache Velocity Engine to execute **data and schema transformations**

- https://github.com/cefriel/mapping-template
- Defines a **Mapping Template Language (MTL)** to enable the description of mapping rules based on the data frame abstraction
- Provides a ***Reader*** and ***Formatter*** interfaces to support multiple input/output.
- Currently interfaces for to extract data frames from **CSV**, **JSON**, **XML** and **SQL** (MySQL and Postgres) inputs are implemented.
- Available as a library on Maven Central or as a standalone JAR executable via CLI.



C

mapping-template example

Map data from XML input to RDF
(example from the RML specification)

- Extract data frame considering all the required elements
- Iterate over the data frame defining the RDF triples to be materialised

```
#set($stops = $reader.getDataframe("
    for $stop in /transport/bus/route//stop
    return map {
        "stopId": $stop/@id,
        "stopName": $stop/text(),
        "busId": $stop/ancestor::bus/@id
}"))

#foreach($stop in $stops)
ex:$stop.busId a transit:Stop ;
  transit:stop "$stop.stopId"^^xsd:int ;
  rdfs:label "$stop.stopName" .
#end
```

a

```
<#TransportMapping> a rr:TriplesMap;
  rml:logicalSource [
    rml:source "Transport.xml" ;
    rml:iterator "/transport/bus";
    rml:referenceFormulation ql:XPath;
  ];

  rr:subjectMap [
    rr:template "http://trans.example.com/{@id}";
    rr:class transit:Stop
  ];

  rr:predicateObjectMap [
    rr:predicate transit:stop;
    rr:objectMap [
      rml:reference "route/stop/@id";
      rr:datatype xsd:int
    ]
  ];

  rr:predicateObjectMap [
    rr:predicate rdfs:label;
    rr:objectMap [
      rml:reference "route/stop"
    ]
  ].
```

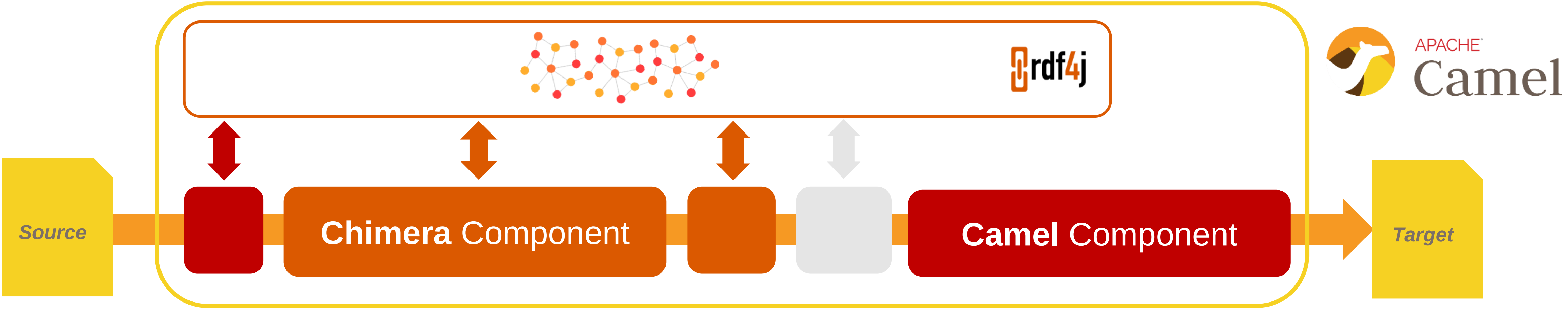
b

mapping-template and Chimera

mapping-template integrated within the Chimera [4] framework (https://github.com/cefriel/chimera) as **Mapping Template Component** to enable the definition of semantic conversion pipelines

Design decision: Limit the dependencies within the mapping-template related to data IO

- MTL currently supports *Data Source* and *Data Sink Specification* from **local file or remote DB** (SQL, RDF)
- The declarative specification of the data source/sink can happen within a Chimera pipeline using the **Camel DSL** → we plan to investigate the support w.r.t RML-IO



[4] M. Grassi, M. Scrocca, A. Carenini, M. Comerio, I. Celino, **Composable Semantic Data Transformation Pipelines with Chimera**, KGCW 2023, https://ceur-ws.org/Vol-3471/paper9.pdf

Qualitative Evaluation

- Several **example mapping templates** made available online in the tool repository considering R2RML/RML mappings discussed in the different specifications

- Qualitative evaluation considering the ontological requirements of the **Conceptual Mapping ontology**
  - 20 requirements fully covered
  - 8 indirectly or partially

(+) Less verbose especially (e.g., RDF-star example in the image)
(+) Flexible textual output enables different types of transformations (e.g., RDF → RDF, CSV → JSON, etc.)
(-) Not fully declarative specification

```
#set ($data = $reader.getDataframe())

#foreach($row in $data)
    << << ex:$row.entity a ex:$row.type >>
        ex:confidence $row.confidence >>
        ex:predictedBy ex:$row.predictor .
#end
```

c

d

```
<#innerTriplesMap>
    a rml:NonAssertedTriplesMap;
    rml:logicalSource ex:PredictionsSource;
    rml:subjectMap [
        rml:template "http://example.com/{entity}";
    ];
    rml:predicateObjectMap [
        rml:predicate rdf:type;
        rml:objectMap [ rml:template "http://example.com/{class}" ];
    ].

<#middleTriplesMap>
    a rml:NonAssertedTriplesMap;
    rml:logicalSource ex:PredictionsSource;
    rml:subjectMap [
        rml:quotedTriplesMap <#innerTriplesMap>;
    ];
    rml:predicateObjectMap [
        rml:predicate ex:confidence;
        rml:objectMap [ rml:reference "confidence" ];
    ].

<#outerTriplesMap>
    a rml:AssertedTriplesMap;
    rml:logicalSource ex:PredictionsSource;
    rml:subjectMap [
        rml:quotedTriplesMap <#middleTriplesMap>;
    ];
    rml:predicateObjectMap [
        rml:predicate ex:predictedBy;
        rml:objectMap [ rml:template "http://example.com/{predictor}" ];
    ].
```

Quantitative Evaluation

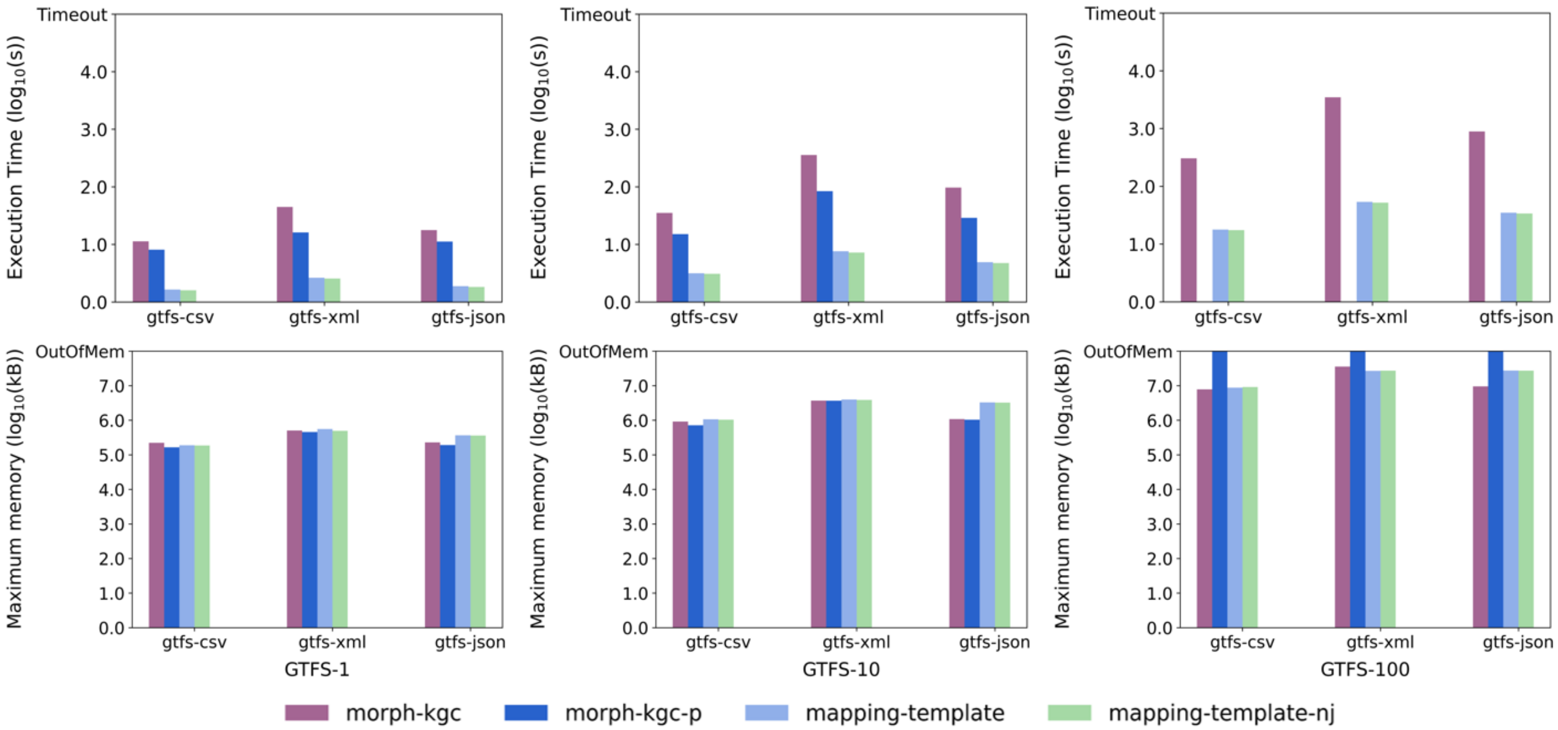
Mapping-template tested vs *morph-kgc* on the GTFS Madrid Benchmark. Same KG construction task but using MTL and RML mappings. MTL mappings manually defined.

(+) Good performance results w.r.t. morph-kgc on the same KG construction task

(+) *–nj* (no-join template using "URI matching") does not affect the performance

(-) mapping-template generates a "textual output" not checking the presence of duplicates (morph-kgc does this by default)

(-) check behaviour of the mapping-template with different types of mappings and against other engines once able to process RML mappings



Conclusions

- The proposed **workflow**:
  - is defined considering the work done by the **KG construction community** and existing solutions
  - supports **declarative knowledge conversion** between different data representations

- The **mapping-template** offers a **tool** implementing the workflow that:
  - Enables non-RDF output (any textual-based format)
  - Facilitates users not expert with RDF-based specifications in the definition of the mapping rules
  - Provides flexibility to the user in optimizing mapping rules according to the considered mapping scenario

- **Next steps**
  - Enable the **execution of RML mappings** via the mapping-template + detailed **performance comparison**
  - Improve the **MTL specification** to reduce the binding to VTL

Thank you for your attention!

https://github.com/cefriel/mapping-template

Any questions? Write to us or open an issue on Github!

MARIO SCROCCA
Knowledge Engineer
Cefriel

marioscrocca

@mario_scrock

mario.scrocca@cefriel.com

One more thing… Q&A from reviews!

- **Using a template language means that the mapping rules define via MTL are not fully declarative**

Yes, the approach based on templates is not fully-declarative. However, the idea behind MTL is exactly to “limit” the expressiveness of the template language by “following” the defined workflow based on declarative languages for KG construction. We are working to improve this aspect (e.g., declarative *join* as in RML) while also implementing an RML compiler to convert RML mappings to MTL.

- **Introducing optimisations in MTL rules is like writing a custom script for a specific mapping**

No, the type of optimizations referred by the paper are the same ones enabled by the RML Logical Views, e.g., the possibility of defining "tabular views" to improve the data access to the input data according to the mapping rules to be executed. My understanding is that the RML Logical Views proposal is aligned with the proposed workflow (Logical View ←→ Data Frame). The paper RML-view-to-CSV discusses in detail the advantages of the decoupling considering optimizations on RML mappings enabled by the explicit definition of an intermediate tabular data structure.

- **How is the MTL-dependency different from the RDF-dependency?**

The main point discussed in the paper is that the textual output generated via MTL can be defined without requiring a specific syntax. Only the “target” format/schema should be known. For example, to generate RDF-star a user knowing MTL should only be able to write RDF-star, while a user knowing RML should learn how to use RML-star.

One more thing… Q&A from reviews!

- **RDF is still needed to generate RDF triples**

Yes, this is a very good point. However, our experience is that users using MTL and not knowing RDF can be provided with “samples” of the target RDF to be used to define mapping rules. The usage of RDF-based mapping languages requires instead a longer training period.

- **MTL lowers the abstraction level, increases the cognitive complexity and, as a consequence, delegates part of the effort to the users. While this could be appealing for developers, it can be more complicated for non-experts users which will need to familiarize themselves with control structures and their logic.**

I really liked this comment, and we will for sure take this aspect into account to perform a user evaluation on the usage of MTL. Our intuition is that while it is probably true that MTL can be more appealing for developers, a non-expert users may in any case be facilitated by reasoning on a “data frame abstraction” instead of dealing with triple maps.