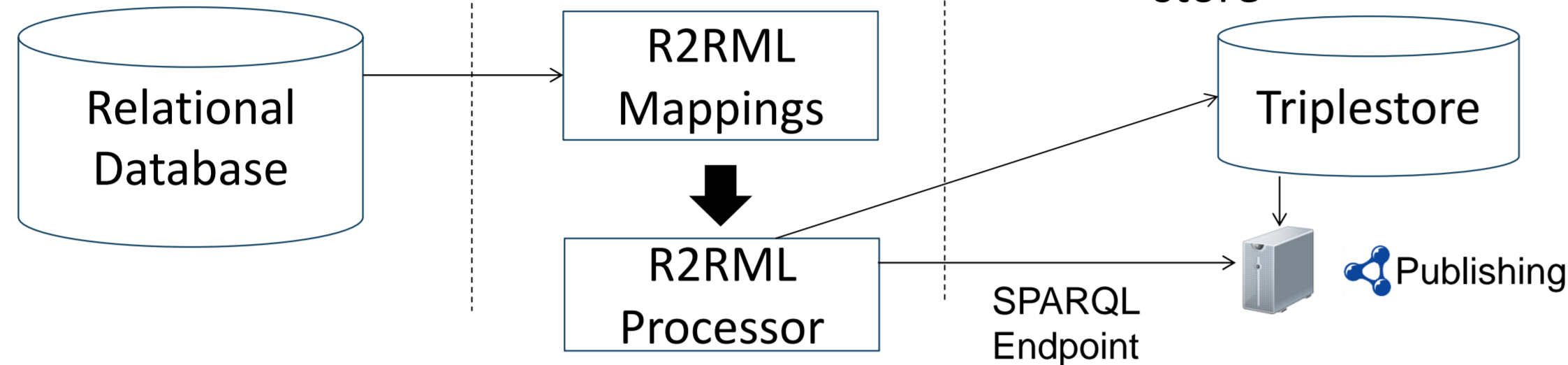


## Overview

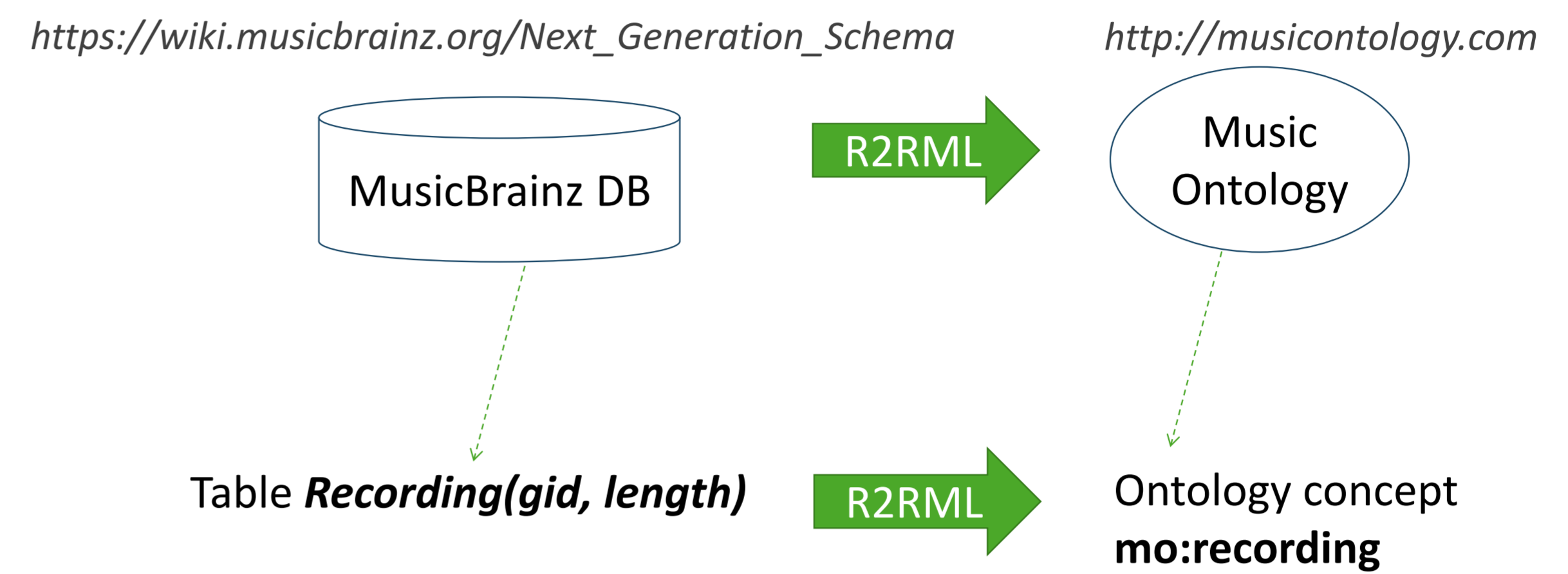
The new W3C standard R2RML defines a language for expressing mappings from relational databases to RDF, allowing applications built on top of the W3C Semantic Technology stack to seamlessly integrate relational data. A major obstacle in using R2RML, though, is the creation and maintenance of mappings. In this demo, we present a novel R2RML mapping editor which provides a user interface to create and edit mappings interactively even for non-experts.

## R2RML

- Task:** Integrate data from relational DBMS with Linked Data
- Approach:** Map from relational schema to semantic vocabulary with R2RML
- Publishing:** Two alternatives
  1. Translate SPARQL into SQL on the fly
  2. Batch transform data into RDF, index and provide SPARQL access in a triple store



## Use Case Example



## Resulting TriplesMap:

```
rr:logicalTable
  [rr:sqlQuery
    """"SELECT id, gid, length FROM musicbrainz.recording
    WHERE musicbrainz.recording.length IS NOT NULL""""]
rr:subjectMap
  [rr:class mo:recording ;
    rr:template "http://musicbrainz.org/artist/{gid}#_" ]
rr:predicateObjectMap
  [rr:objectMap
    [ rr:column "length" ;
      rr:datatype <http://www.w3.org/2001/XMLSchema#float > ;
      rr:predicate mo:duration ] ;
```

That is already too complex!

## Our Solution: A New R2RML Mapping Editor

- Features:**
- A solution to create, edit, manage R2RML mapping rules easily.
  - Easy to use user interface to create mappings.
  - Import/Export mappings from the editor directly.
  - Preview generated triples at various steps of mapping creation.
  - Mappings can be created without learning the R2RML vocabulary.
  - Integrated with the Information Workbench platform.



## Steps of the R2RML mapping rule management using the editor

- **Datasource, Base URI Selection:** In the first step the user can select from all the available datasources or configure a new one. The user also has the option to configure the base URI for template generation.
- **R2RML Rules:** At this step all the rules are displayed. The editor provides many options to browse the existing rules. E.g. filter the rules by table names.

- **Subject Map Creation:** By the click of a button, the user can carry over columns that should be used to generate the subject URI template. Additionally, classes (i.e., types) can be added to the subject by using the auto-complete field rdf:type, which displays all the classes that are present in the system.

**Choose from different modes: Default/ Ontology Statistics/ Tables Statistics.**

**Filter mappings by Table Name, Class Name or Property Name.**

**List of existing mapping rules.**

**Export/Import mapping rules To/From files.**

**Subject Map.**

SQL Preview for: select track\_id as tid, recording\_id as r...

Preview of generated triples:

```
<http://www.fluidops.com/resource/17> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/13684156> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/18> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/13684155> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/10> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/15> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/11> a <http://purl.org/ontology/mot/Track> .
<http://www.fluidops.com/resource/12759010> a <http://purl.org/ontology/mot/Track> .
```

- **Logical Table Selection:** During this step the user chooses a logical table. The editor provides the option of choosing a table or typing a SQL query. For tables all the metadata and data values can be looked up by the user.

- **Predicate-Object Maps Creation:** At this step the user can select the predicate and objects for the triples to be generated. All types of predicates are supported by the editor, e.g. constants, URI templates. All object types are also supported namely: constants, literals from column values, templates, reference to other mappings.

**Action buttons to get the overview of the tables before selecting one.**

**Subject template:** http://www.fluidops.com/resource/{tid}

**predicate:** Constant

**Object Type:** Reference

**Parent Rule:** https://github.com/LinkedData/2to3/blob/...

**Join Condition:** Child Column: gid, Parent Column: gid

- **Textual Representation:** Finally, an intuitive textual representation of the mapping rule is displayed. At this point the user could go back to any previous step to modify the mapping or otherwise save the mapping.

