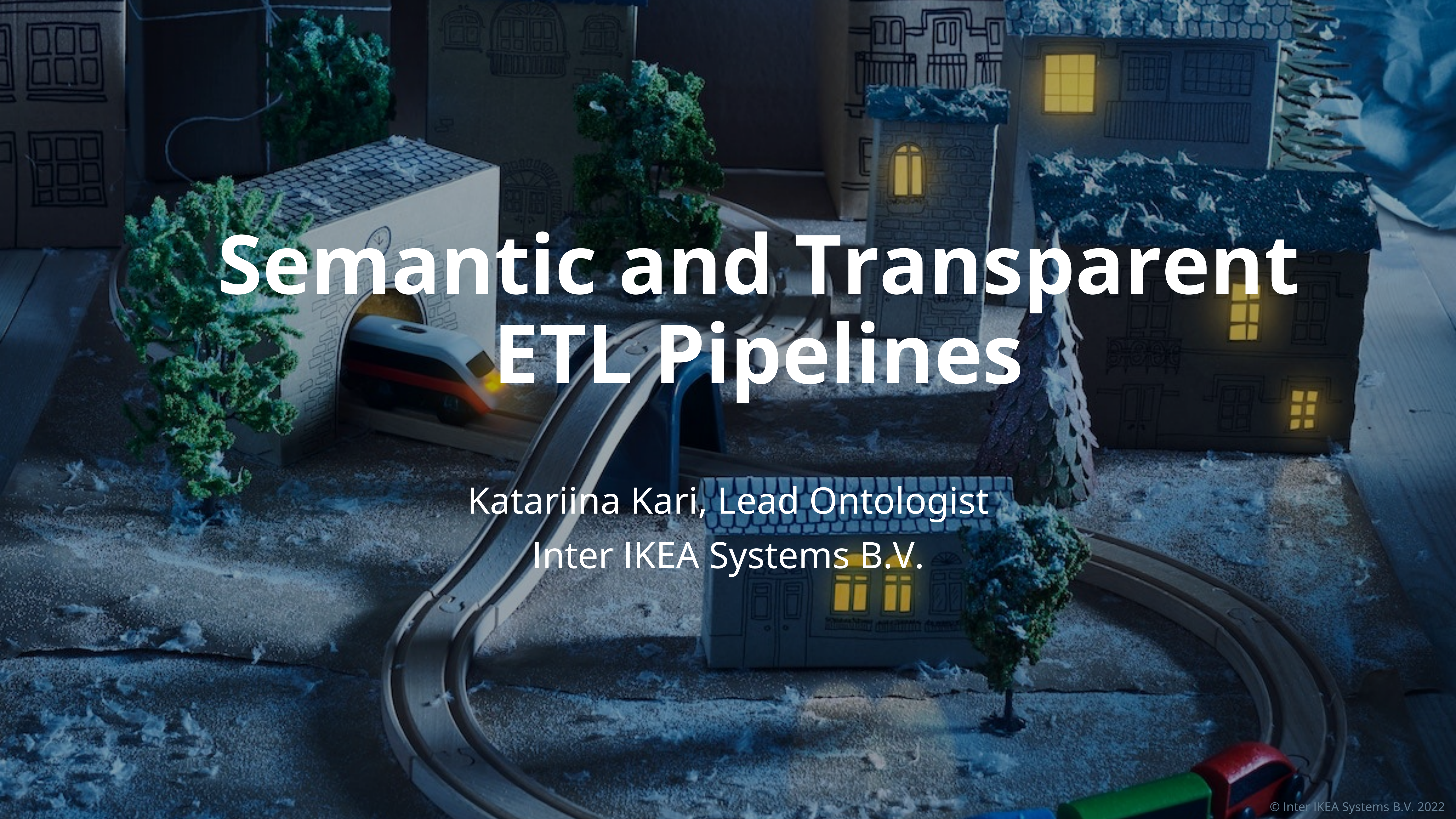


Semantic and Transparent ETL Pipelines

Katariina Kari, Lead Ontologist

Inter IKEA Systems B.V.

Semantic ETL

- Turn non-graph data sources into RDF data graph
- That RDF output conforms to a SHACL definitions of the ontology
- The ontology for the RDF is pre-defined
- Extraction: process data source
- **Transform: create triples**
- Load: upload in graph database

Past experience

- Working with product data at **Zalando** and **IKEA**
- Working on Master Data Management project at Zalando
- Reoccuring enterprise data needs (psst. KGs can solve them!)
  - Data lineage
  - Data catalogue
  - Data governance

→ Data source has IRI, ands its keys has IRIs

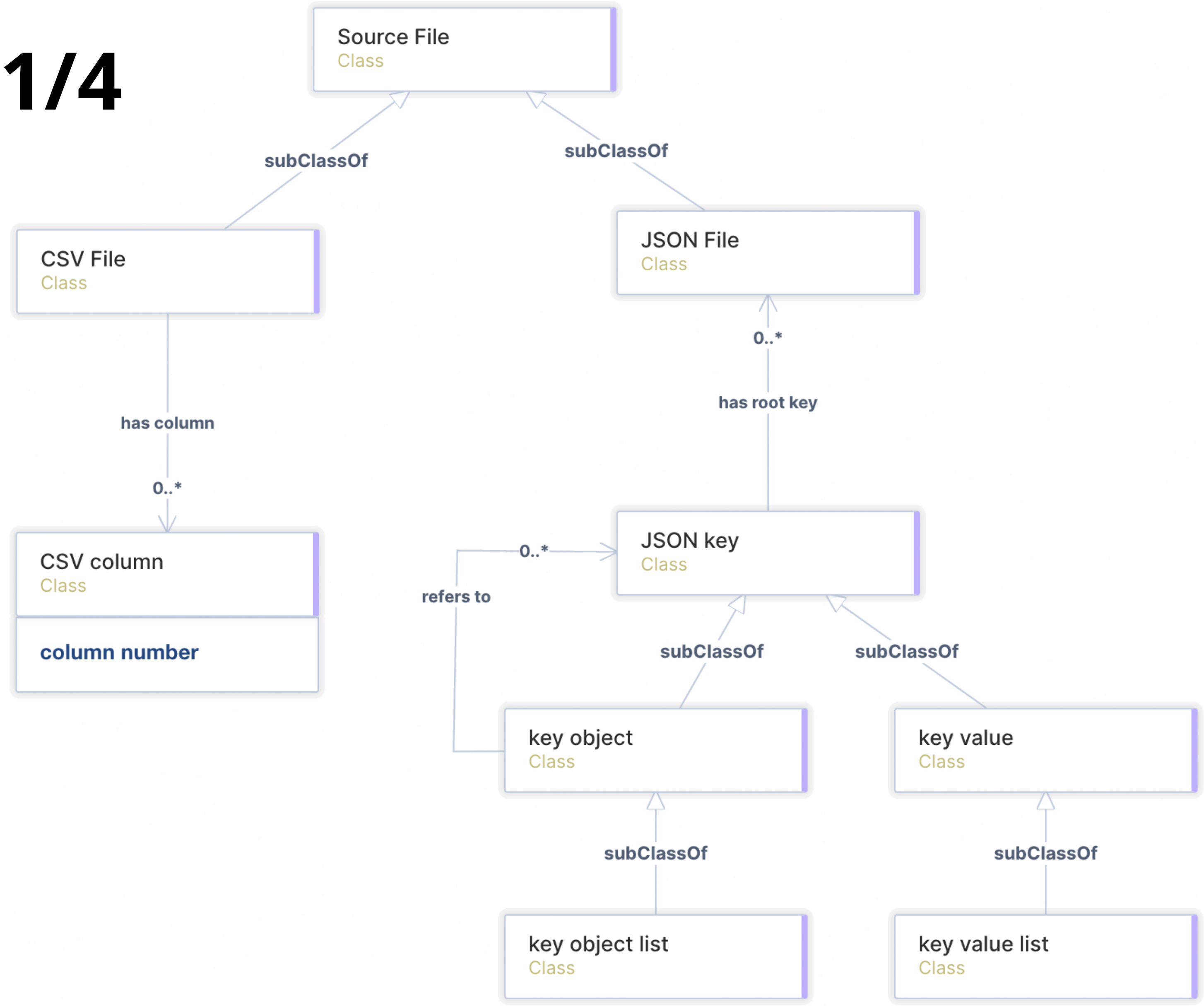
Current Requirements for Semantic ETL

- Enable **any data transformation** as part of the mapping
- Mapping to RDF is intuitive, easy to change, easy to debug
- Mapping supports at least JSON, CSV
- Mapping could be authored by domain experts
- Store code centrally for transparency and better management of changes:
  - Pattern for IRI creation
  - Data transformation functions
- Mapping can return RDF-star triples

Proposed Solution 1/4

Data Source Ontology for describing data sources

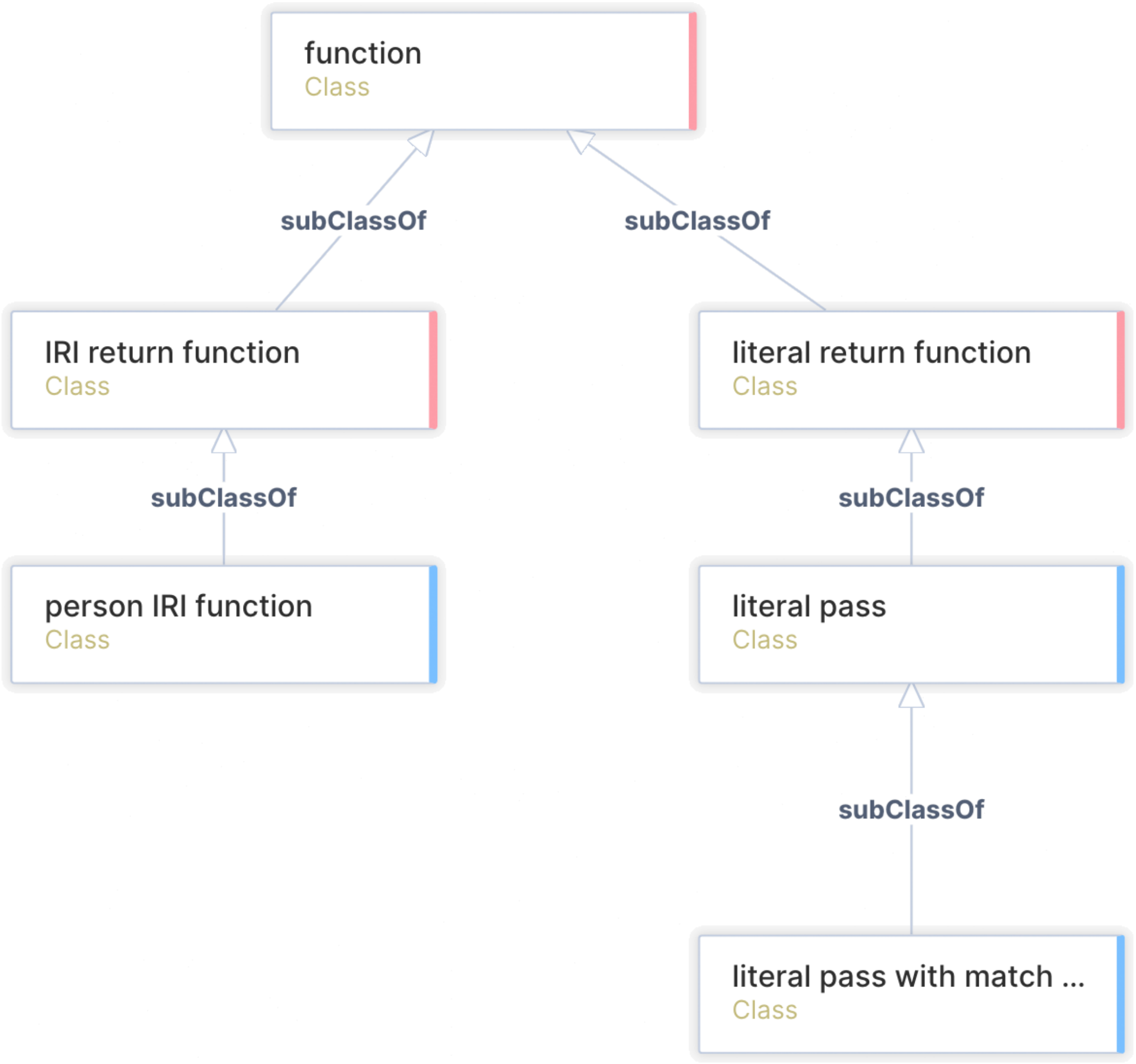
- Each data source is an instance with an IRI
- Each key or column is an instance with an IRI



Proposed Solution 2/4

Functions and Mappings Ontology

- Each function is a class
- Each function parameter is a property
- SHACL constrains properties that are function parameters
- Each mapping is an instance
- Each data transformation is an instance of a function with particular data source keys/columns as its parameters



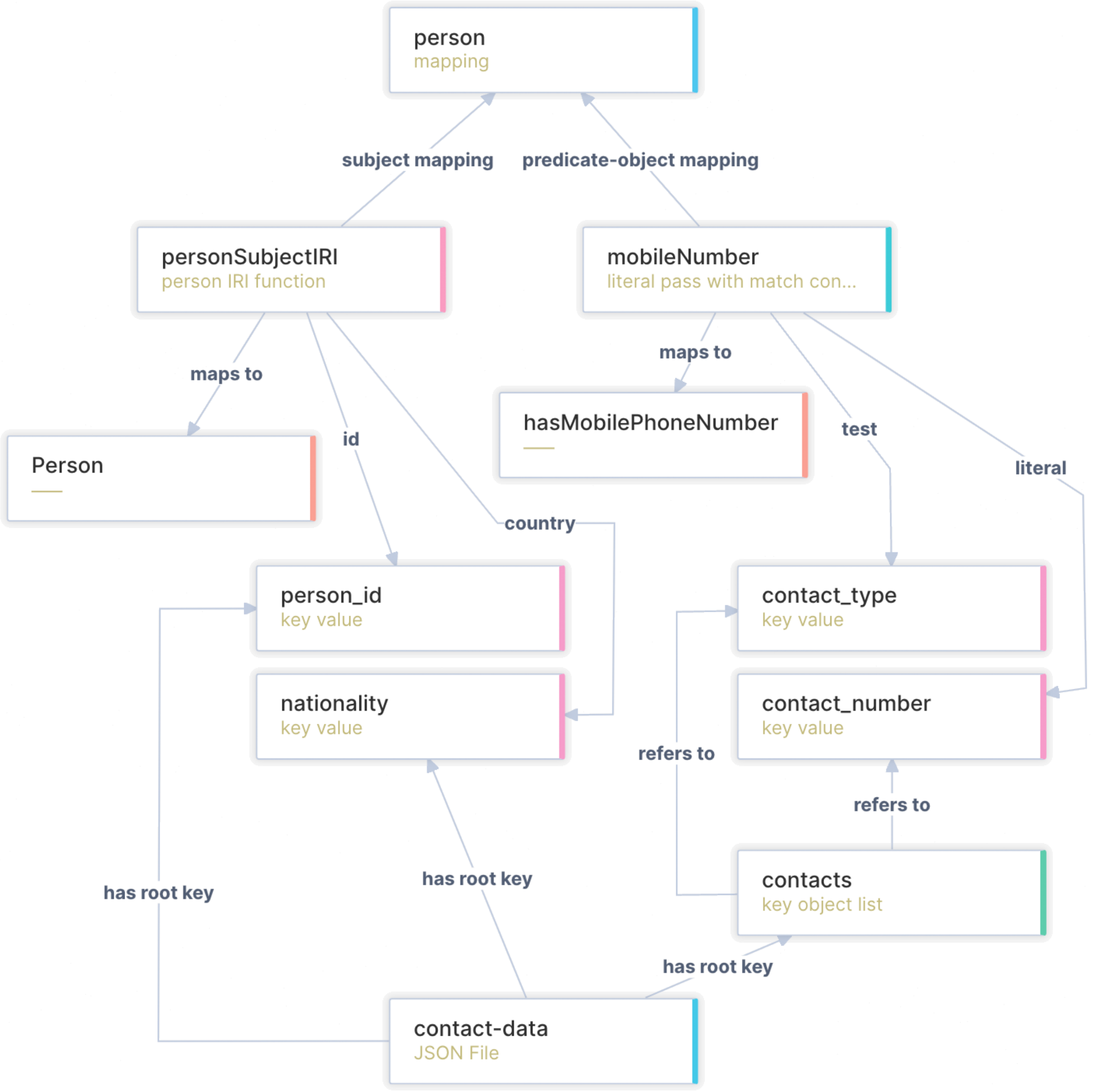
Proposed Solution 3/4

One Mapping Instance

- Produces one or more triples for one subject
- Has one subject mapping that is an instance of its subject IRI function
- Subject mapping creates an rdf:type to its class triple
- Has zero or many predicateObjectMappings that each create one triple

Multiple Mapping instances in one file

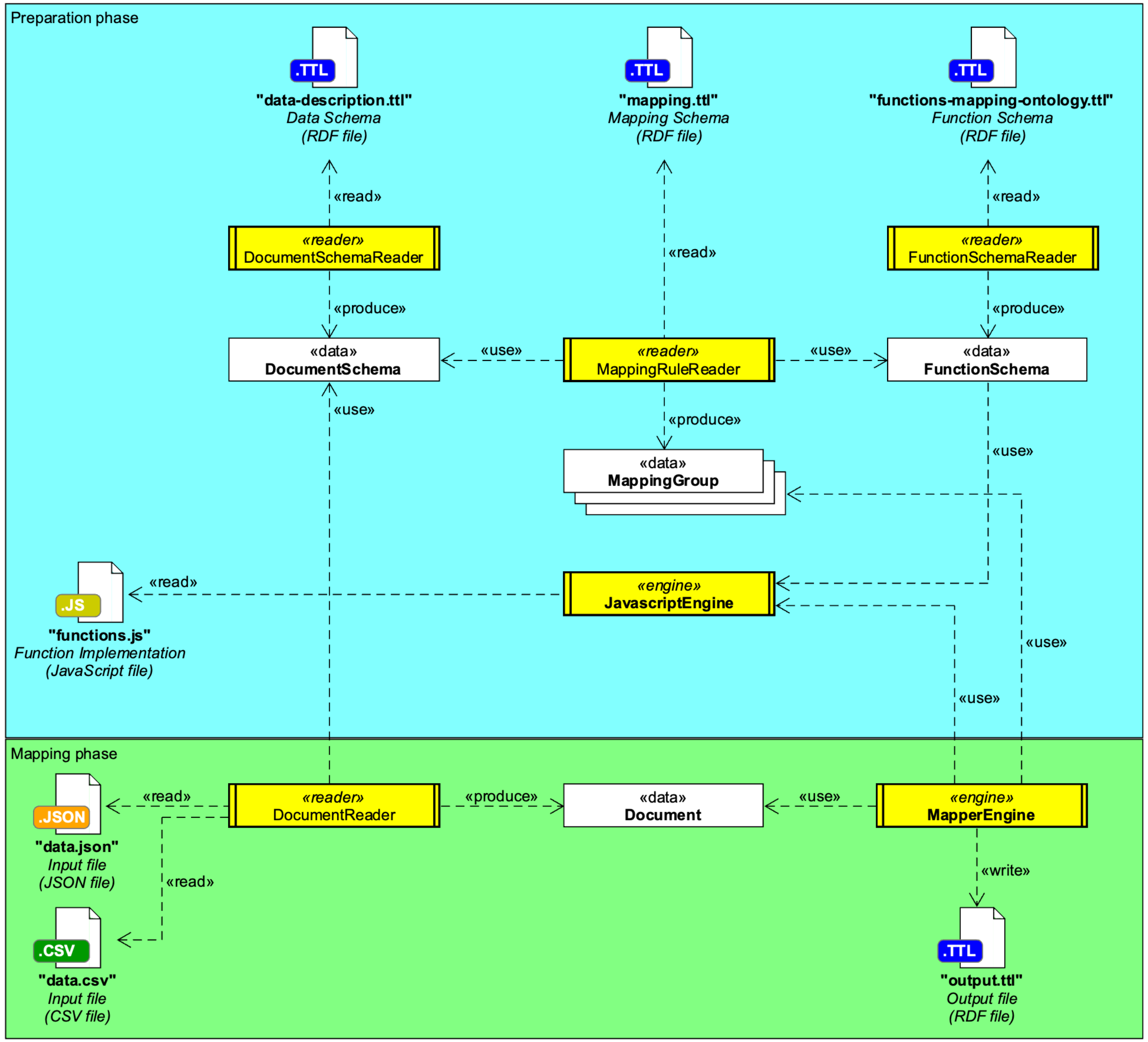
One file per data source



Proposed Solution 4/4

Program that takes in:

- Non-graph data source
- Semantic description of data source
- Function ontology
- Functions.js
- Mapping



Preparation phase

.TTL "data-description.ttl" *Data Schema* *(RDF file)*

.TTL "mapping.ttl" *Mapping Schema* *(RDF file)*

.TTL "functions-mapping-ontology.ttl" *Function Schema* *(RDF file)*

«reader» DocumentSchemaReader

«reader» FunctionSchemaReader

«reader» MappingRuleReader

«data» **DocumentSchema**

«data» **FunctionSchema**

«data» **MappingGroup**

«engine» **JavascriptEngine**

.JS **"functions.js"** *Function Implementation* *(JavaScript file)*

«read» «produce» «use» «read» «use» «write»

Mapping phase

.JSON **"data.json"** *Input file* *(JSON file)*

.CSV **"data.csv"** *Input file* *(CSV file)*

«reader» DocumentReader

«data» **Document**

«engine» **MapperEngine**

.TTL **"output.ttl"** *Output file* *(RDF file)*

Demo

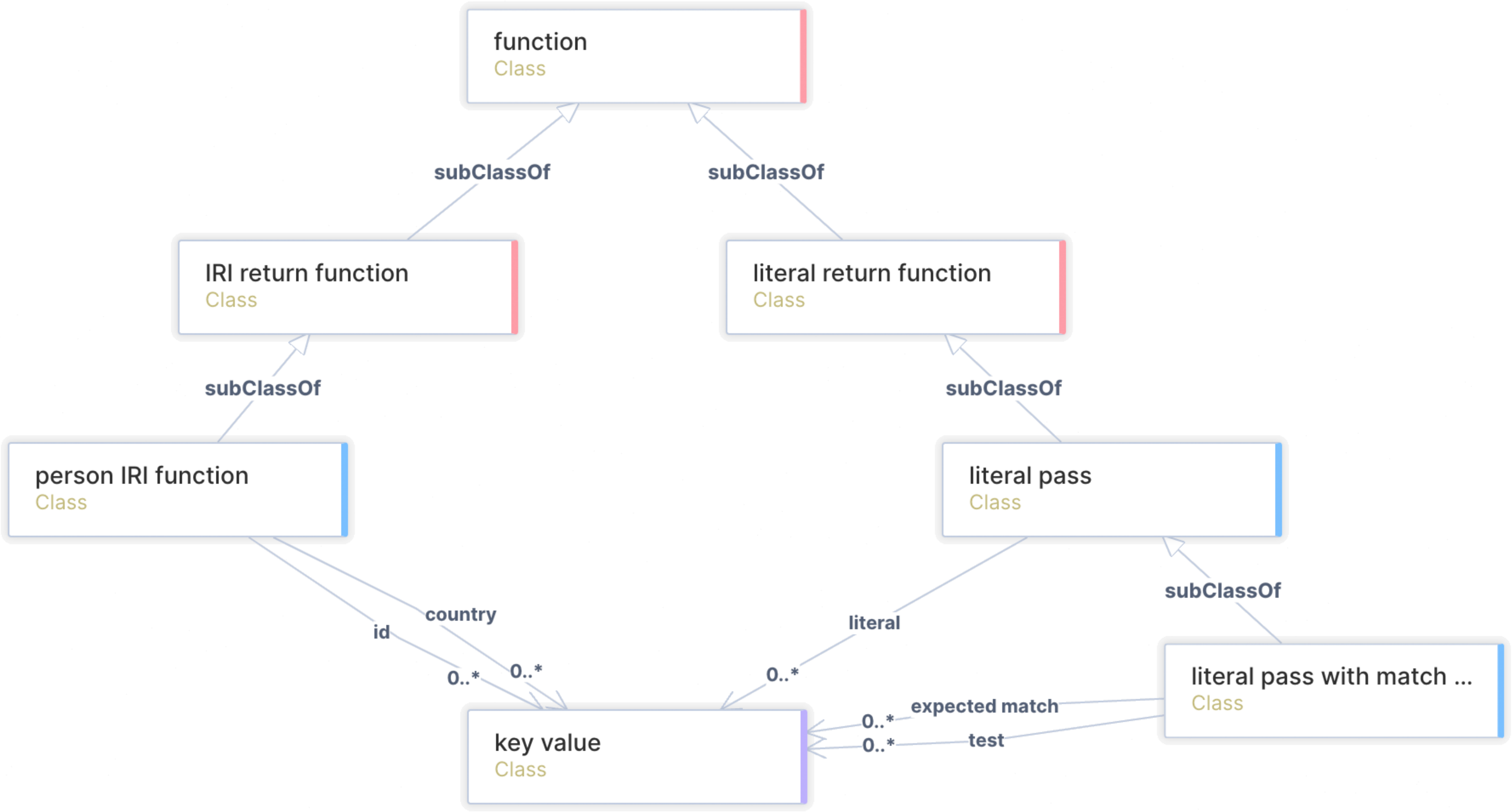
© Inter IKEA Systems B.V. 2022

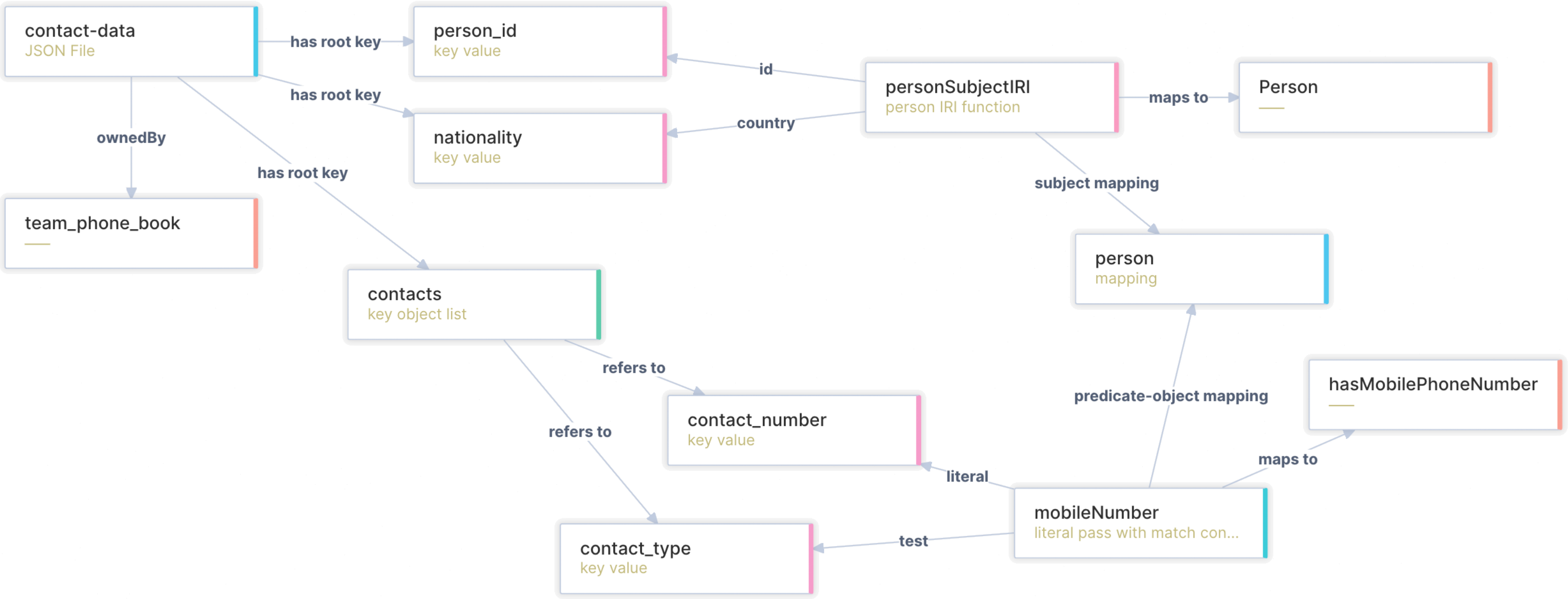
Thank you

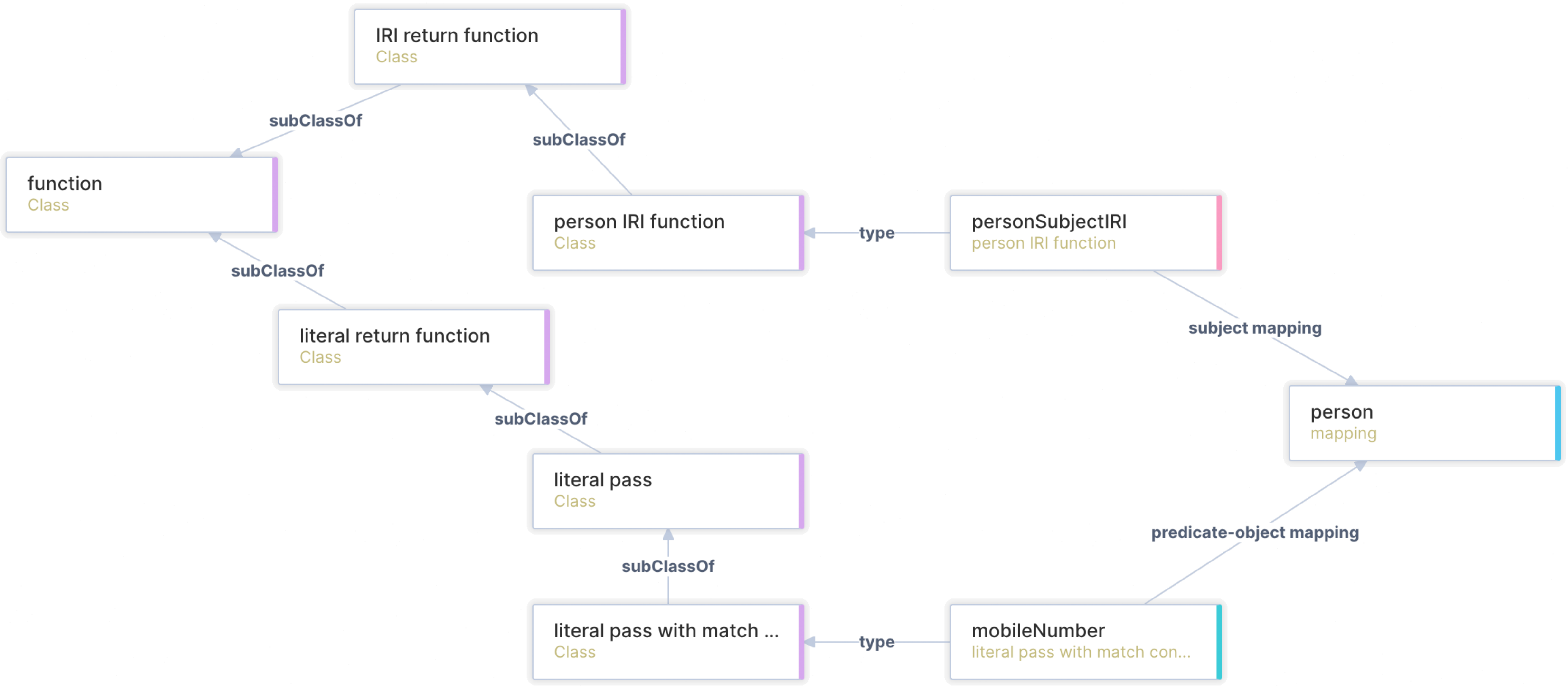
- https://twitter.com/katsi111
- https://github.com/katsi/
- https://www.linkedin.com/in/katsi/

Standards that we considered:

- RML, DCAT, W3C Metadata Vocabulary for Tabular Data, FnO


function
Class
subClassOf
subClassOf
IRI return function
Class
literal return function
Class
subClassOf
subClassOf
person IRI function
Class
literal pass
Class
subClassOf
country
id
literal
literal pass with match ...
Class
0..*
0..*
0..*
expected match
0..*
test
0..*
key value
Class


contact-data
JSON File
has root key
person_id
key value
has root key
nationality
key value
ownedBy
has root key
team_phone_book
id
country
personSubjectIRI
person IRI function
maps to
Person
subject mapping
person
mapping
contacts
key object list
refers to
refers to
contact_number
key value
predicate-object mapping
hasMobilePhoneNumber
maps to
literal
mobileNumber
literal pass with match con...
test
contact_type
key value


IRI return function
Class
subClassOf
subClassOf
function
Class
person IRI function
Class
type
personSubjectIRI
person IRI function
subClassOf
literal return function
Class
subject mapping
person
mapping
subClassOf
literal pass
Class
predicate-object mapping
subClassOf
literal pass with match ...
Class
type
mobileNumber
literal pass with match con...