Bringing SciCat and LinkML together

Linus Pithan (DESY)

July 2024

Bringing SciCat and LinkML together

In analogy to object oriented programming:

- LinkML Schema corresponds to class definitions
- Datasets in SciCat correspond to instances of these classes

What is LinkML?

LinkML is a flexible modeling language that allows you to author schemas in YAML that describe the structure of your data. Additionally, it is a framework for working with and validating data in a variety of formats (JSON, RDF, TSV), with generators for compiling LinkML schemas to other frameworks.

- https://linkml.io
- https://github.com/linkml/linkml/
- Python code / Apache license

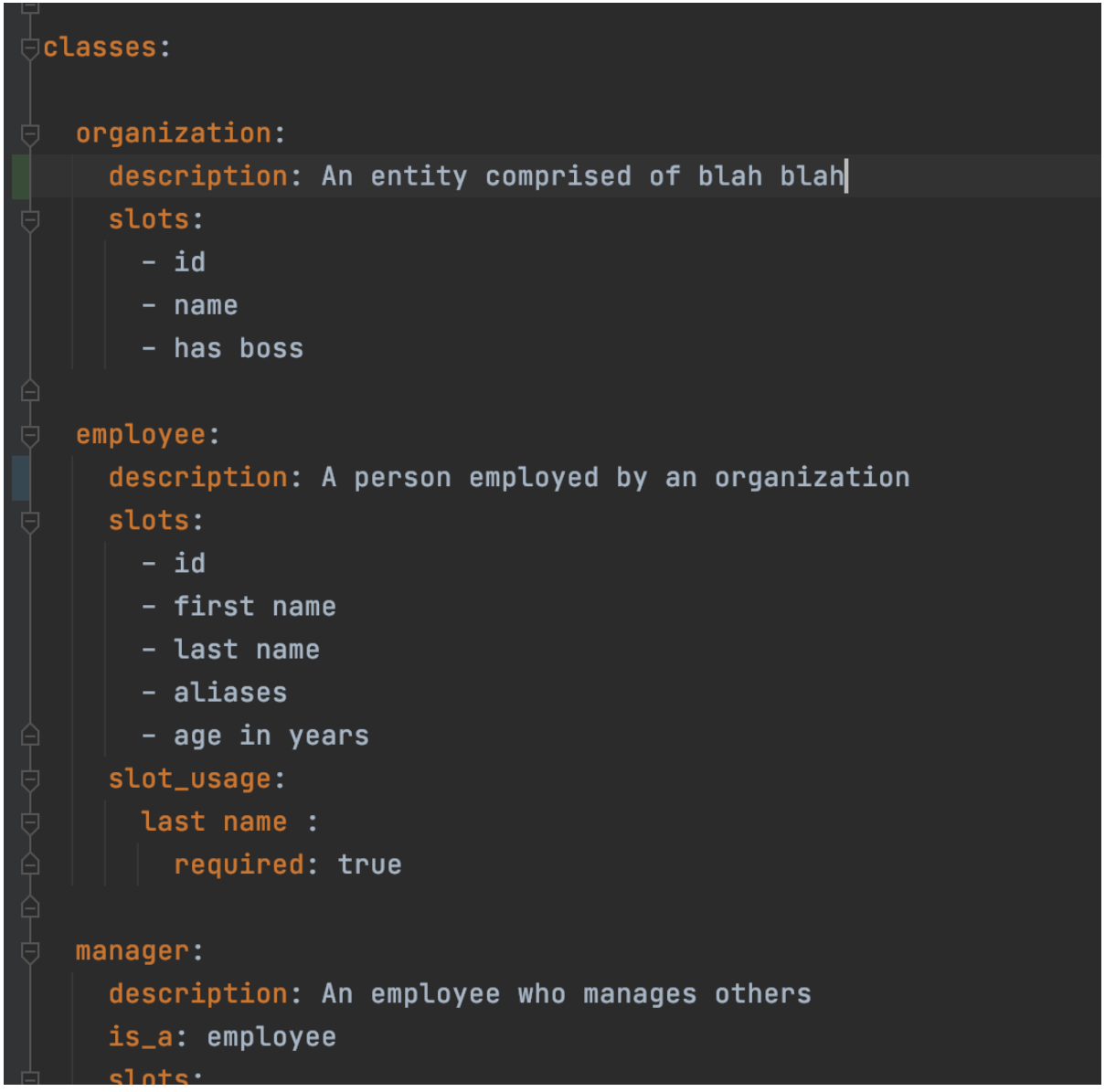
Why are we interested in LinkML

We were looking for a tool that

- could help us define a validatable structure for ScientificMetadata
- would help us to communicate with domain scientists about metadata
- supports mapping to external definitions
- would allow us to share the same definitions with other data catalogs
- supports lots of data description languages
  - specifically it should produce and validate JSON schema
- allows to deposit the data model as git repo

. . . and we found LinkML

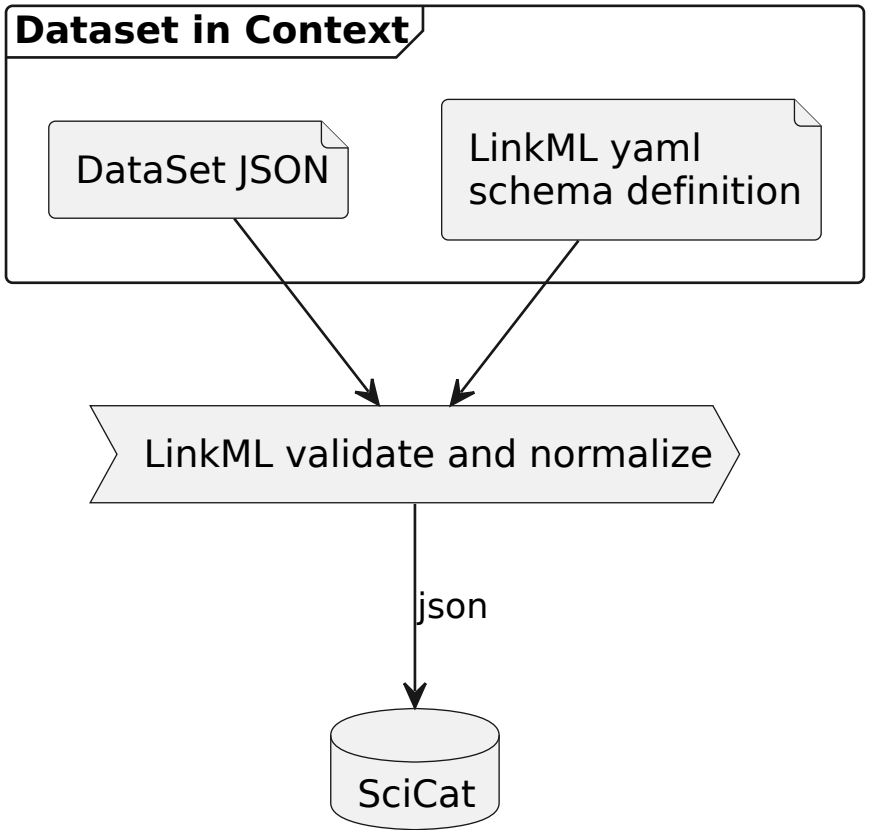
How does a LinkML Schema look like



```yaml
classes:

  organization:
    description: An entity comprised of blah blah
    slots:
      - id
      - name
      - has boss

  employee:
    description: A person employed by an organization
    slots:
      - id
      - first name
      - last name
      - aliases
      - age in years
    slot_usage:
      last name :
        required: true

  manager:
    description: An employee who manages others
    is_a: employee
    slots:
```

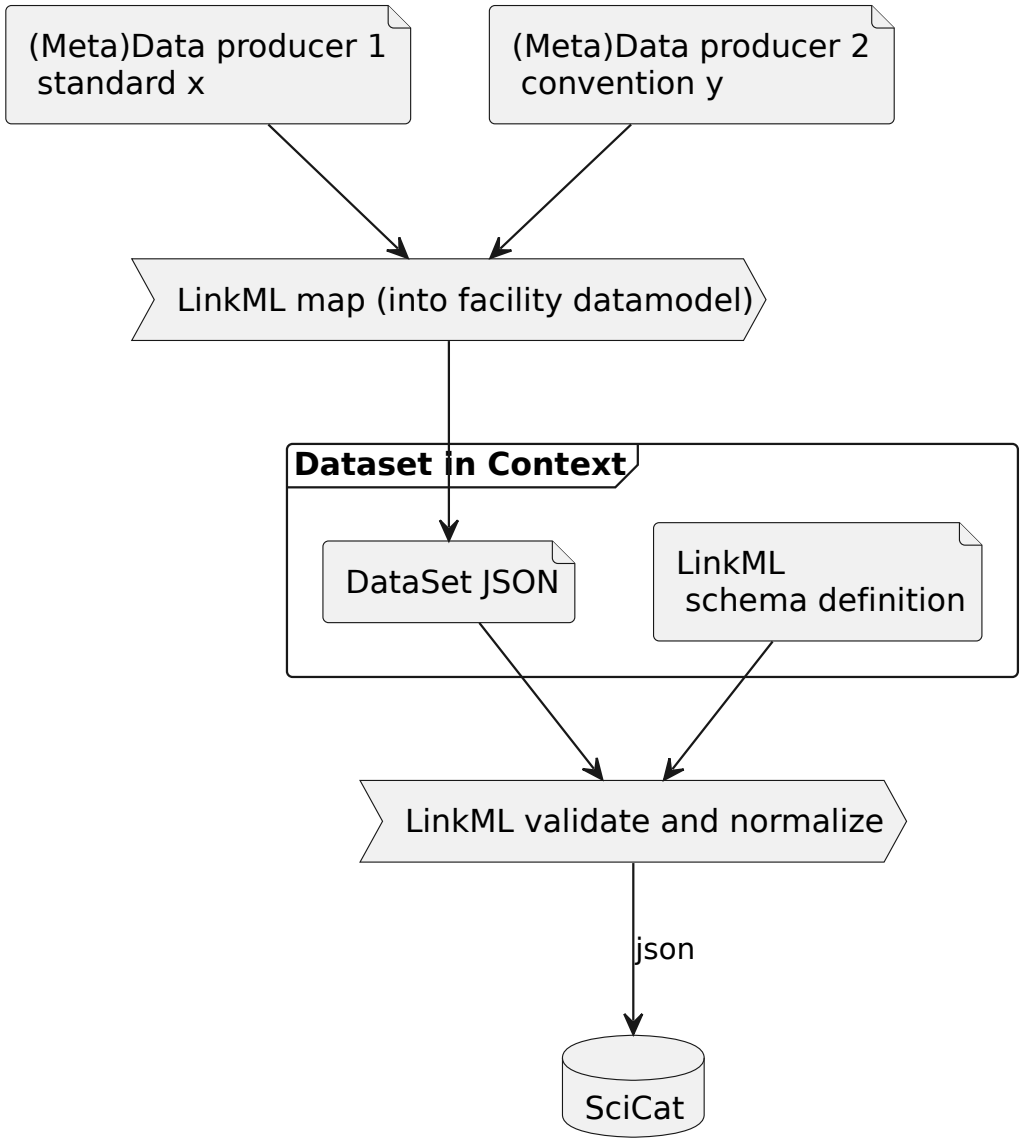
Figure 1: image

Data validation

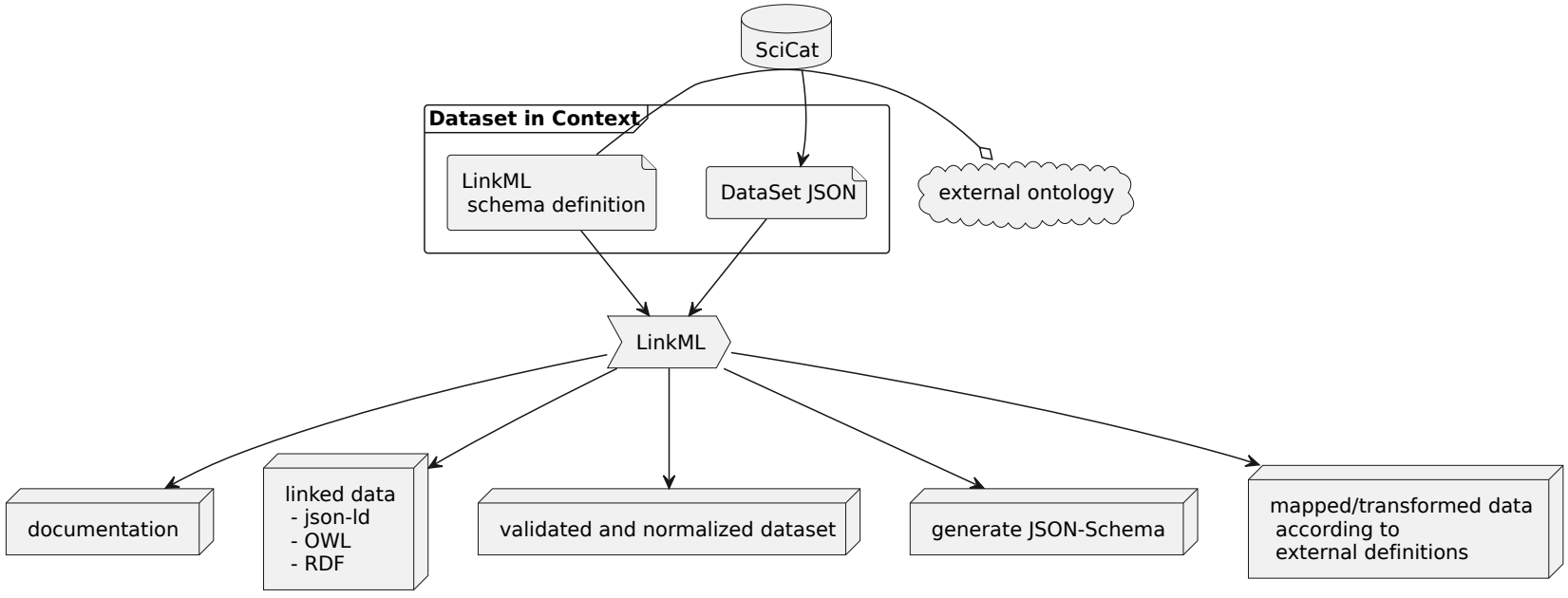


- Provide a validatable data structure for the Scientific Metadata
- impose side-specific rules
- provide a *context*, i.e. reference to external definitions in a machine-actionable way.

Mapping multiple metadata sources into a SciCat dataset



Extracting data from SciCat via LinkML



- Increased consistenty through automatically generated documentation, validated against the datamodel
- Generation of linked data - SciCat only stores *instances* , *class* definitions are kept in LinkML
- Data normalization
- Data transformation

Auto-Generated Documentation

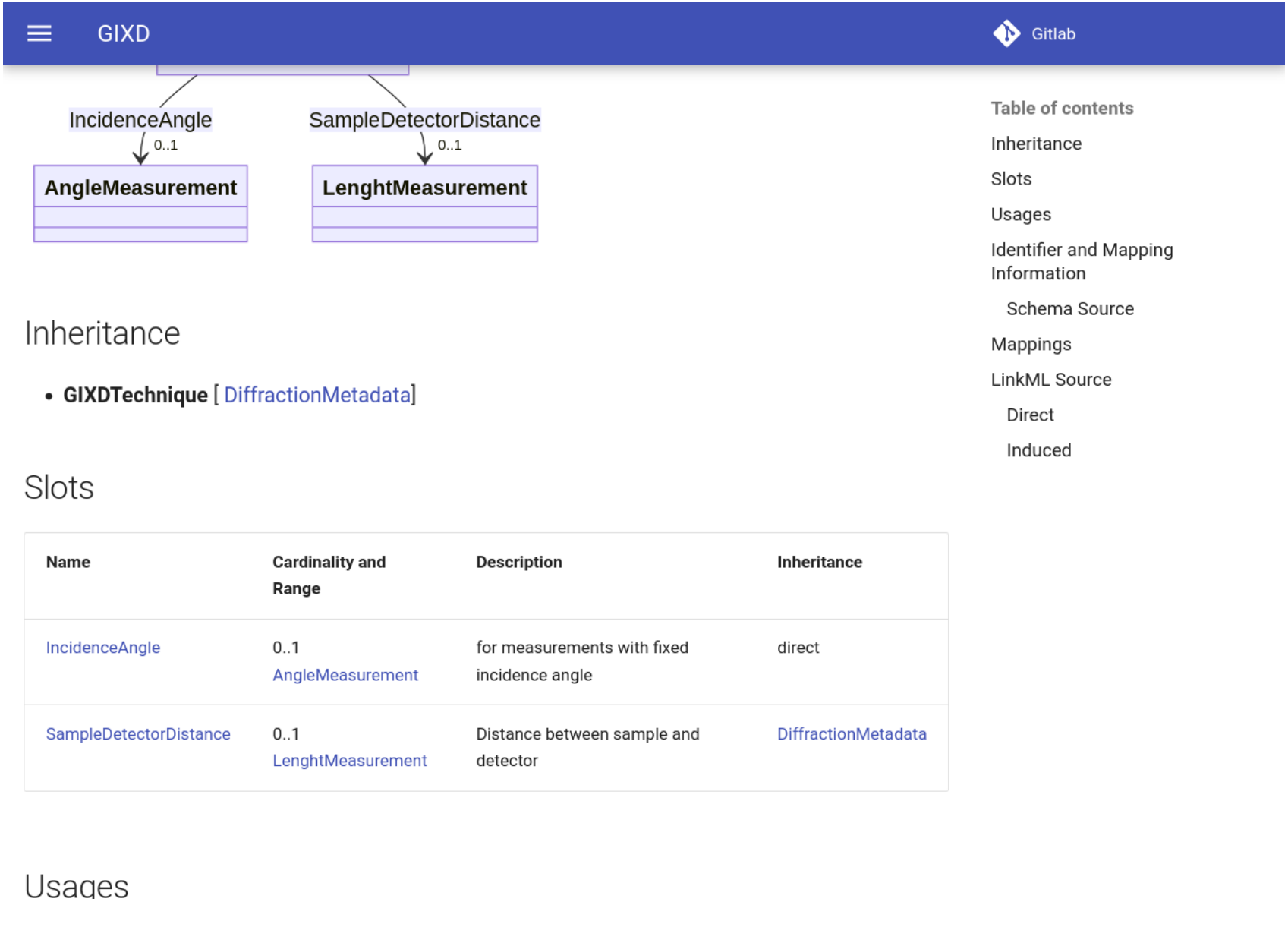
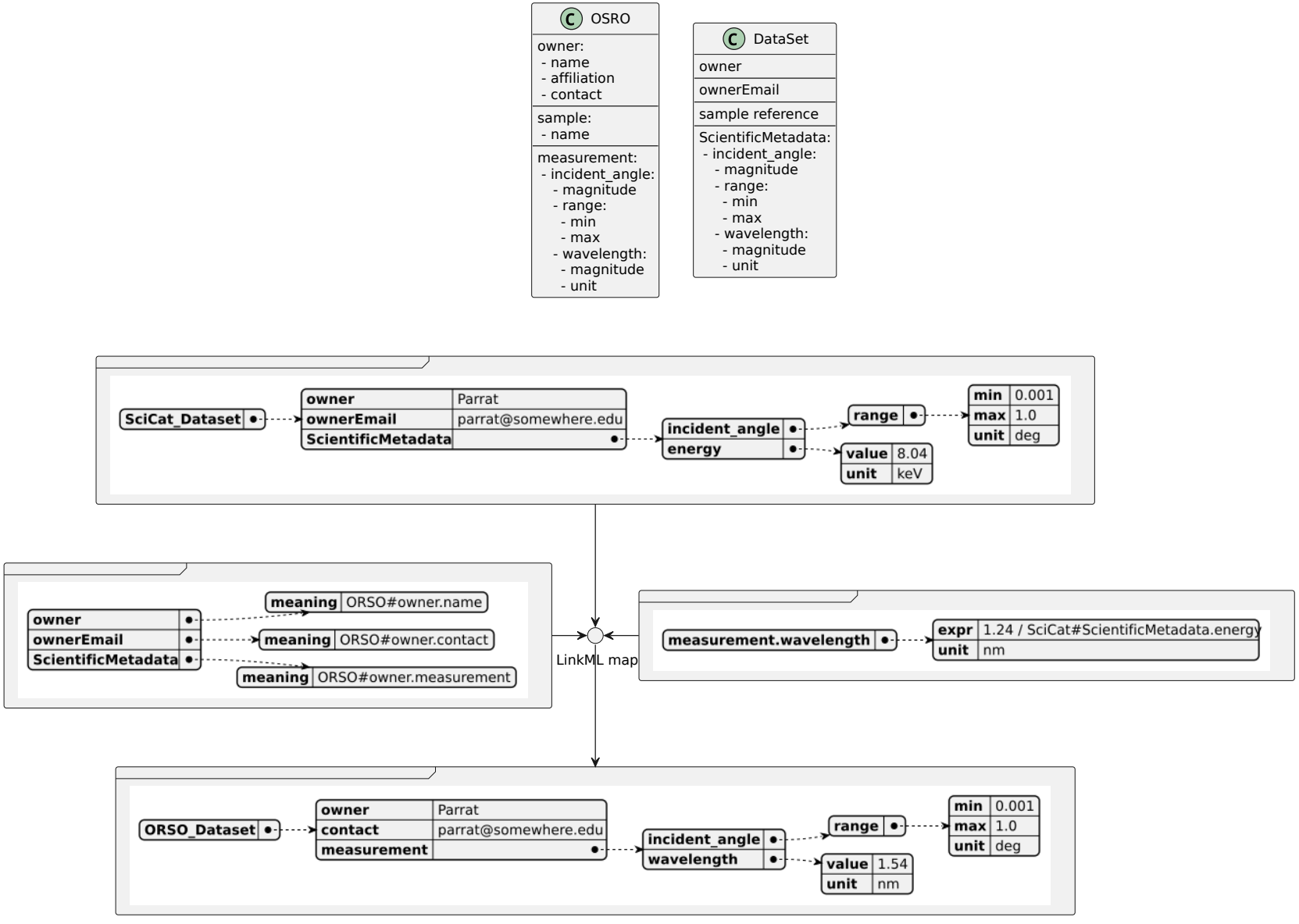
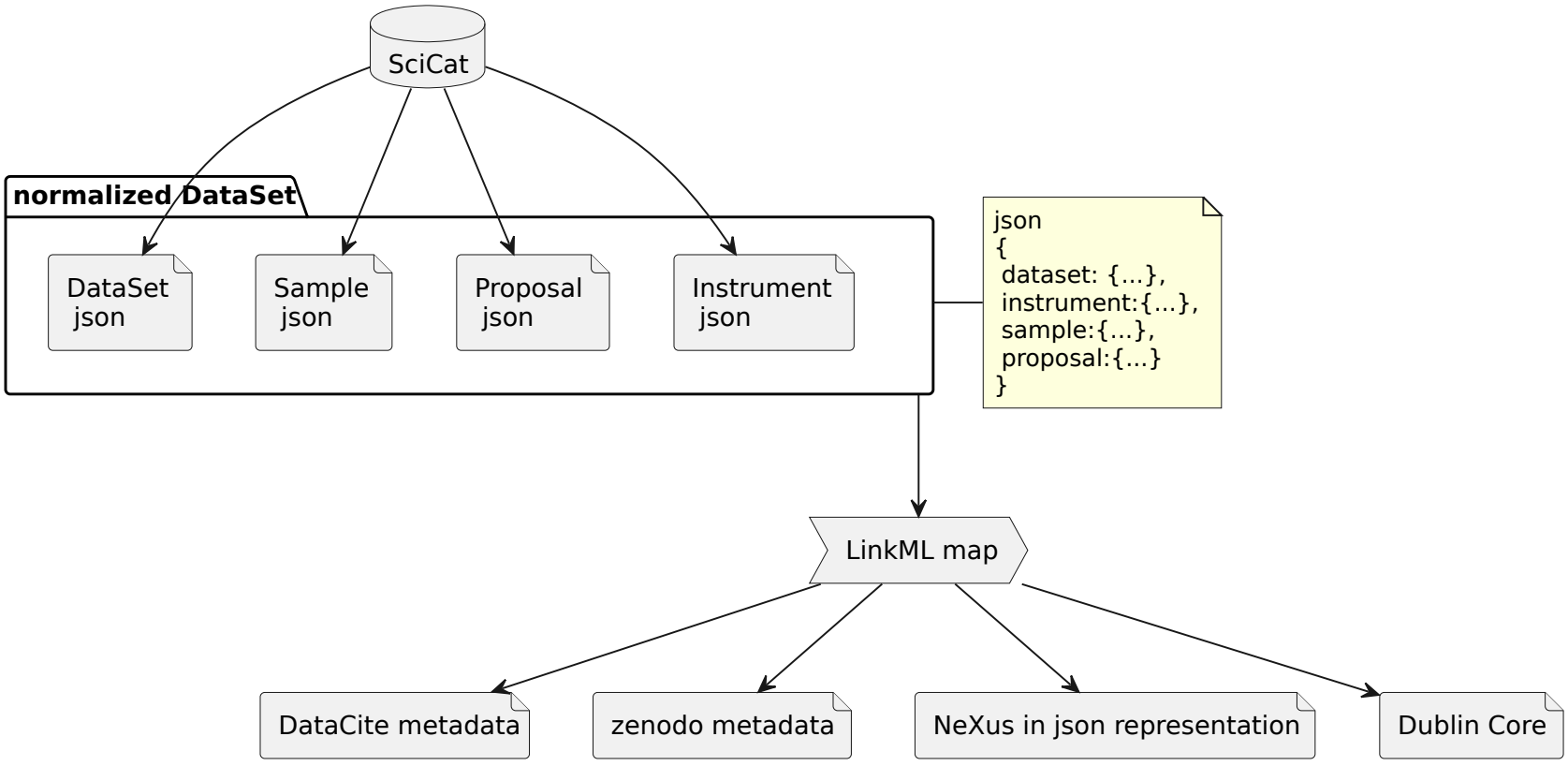


Figure 2: image

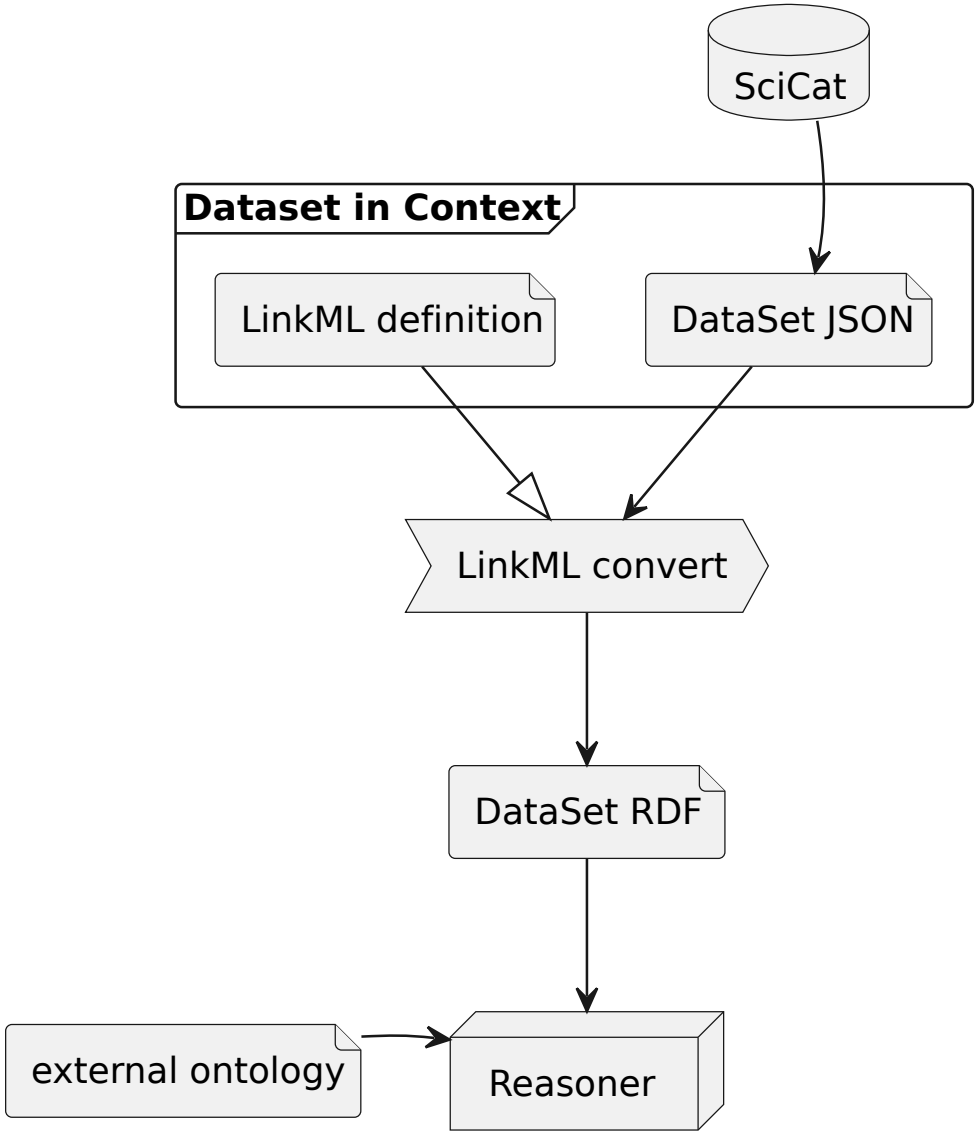
Mapping between SciCat and community standards



converting DataSets including their surrounding



Using DataSets as instances of an ontology



The Pizza ontology

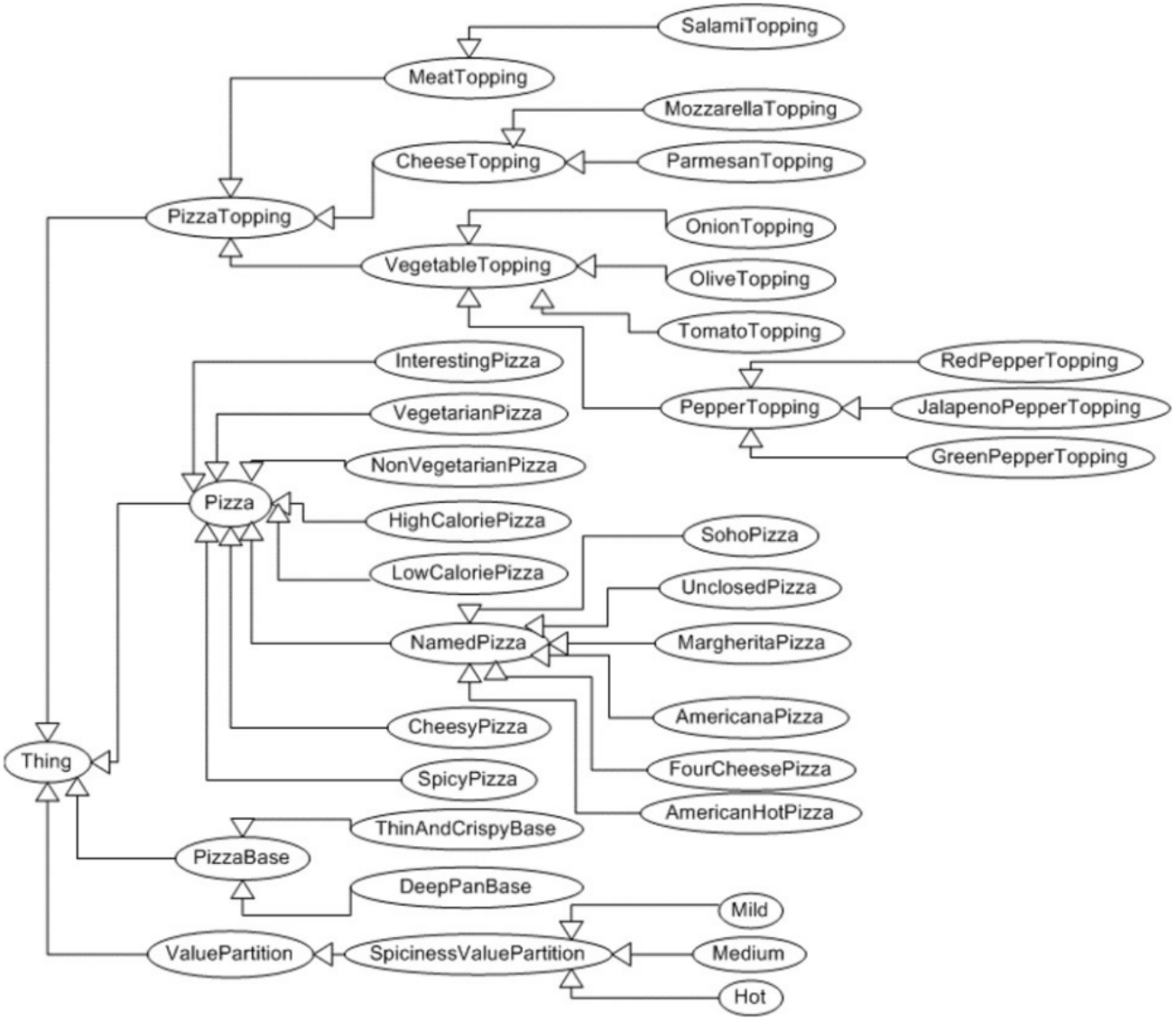


Figure 3: image

A Pizza Dataset in SciCat

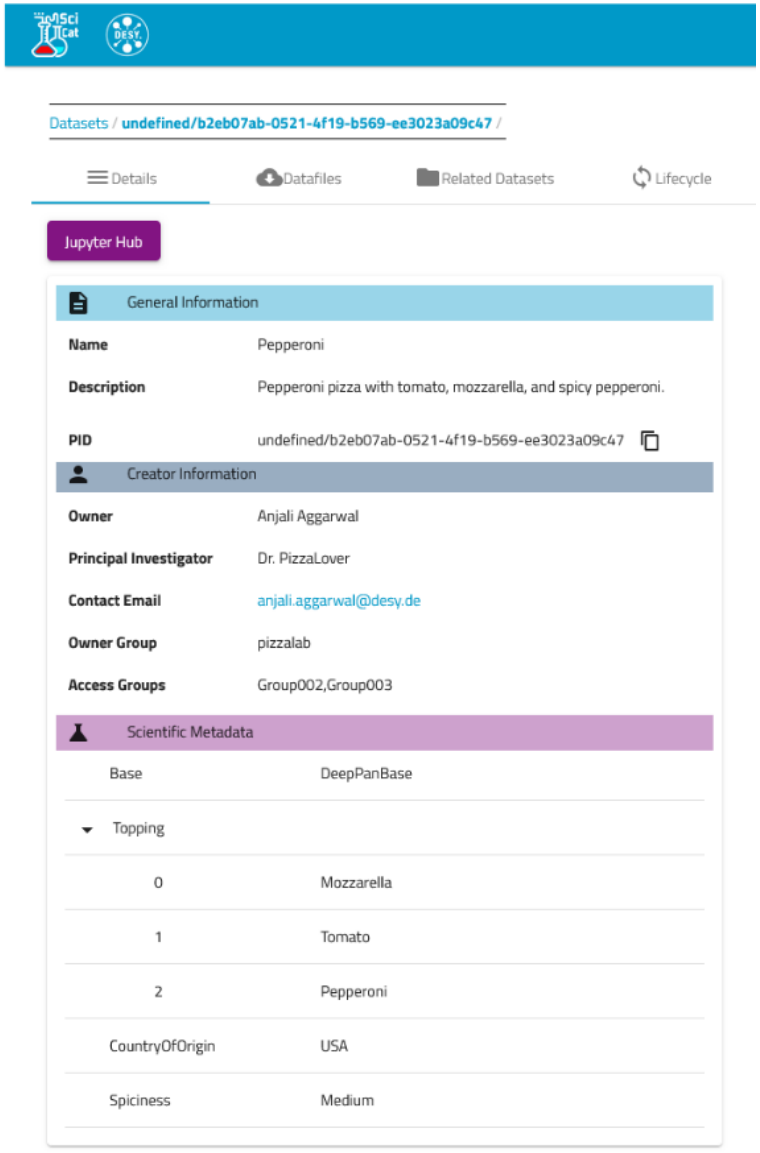


Figure 4: image

How does LinkML interplay with the SciCat datamodel?

pizza.yaml

```
id: http://example.org/pizza
name: Pizza_in_SciCat
description: A schema for validating pizza data
prefixes:
  linkml: https://w3id.org/linkml/
  piz: http://www.co-ode.org/ontologies/pizza

enums:
  PizzaBase:
    permissible_values:
      DeepPanBase:
        meaning: piz:DeepPanBase
        description: Thick Base
      ThinAndCrispyBase:
        meaning: piz:ThinAndCrispyBase
        description: Thin Base
      WholeWheatBase:
        meaning: piz:WholeWheatBase
        description: Thick Base
      StuffedCrustBase:
        meaning: piz:StuffedCrustBase
        description: Thick Base


classes:
  Pizza:
    attributes:
      title:
        range: string
      Base:
        range: PizzaBase
      Topping:
        range: string
        multivalued: true
      CountryOfOrigin:
        range: string
      Spiciness:
        range: SpicinessLevel
```

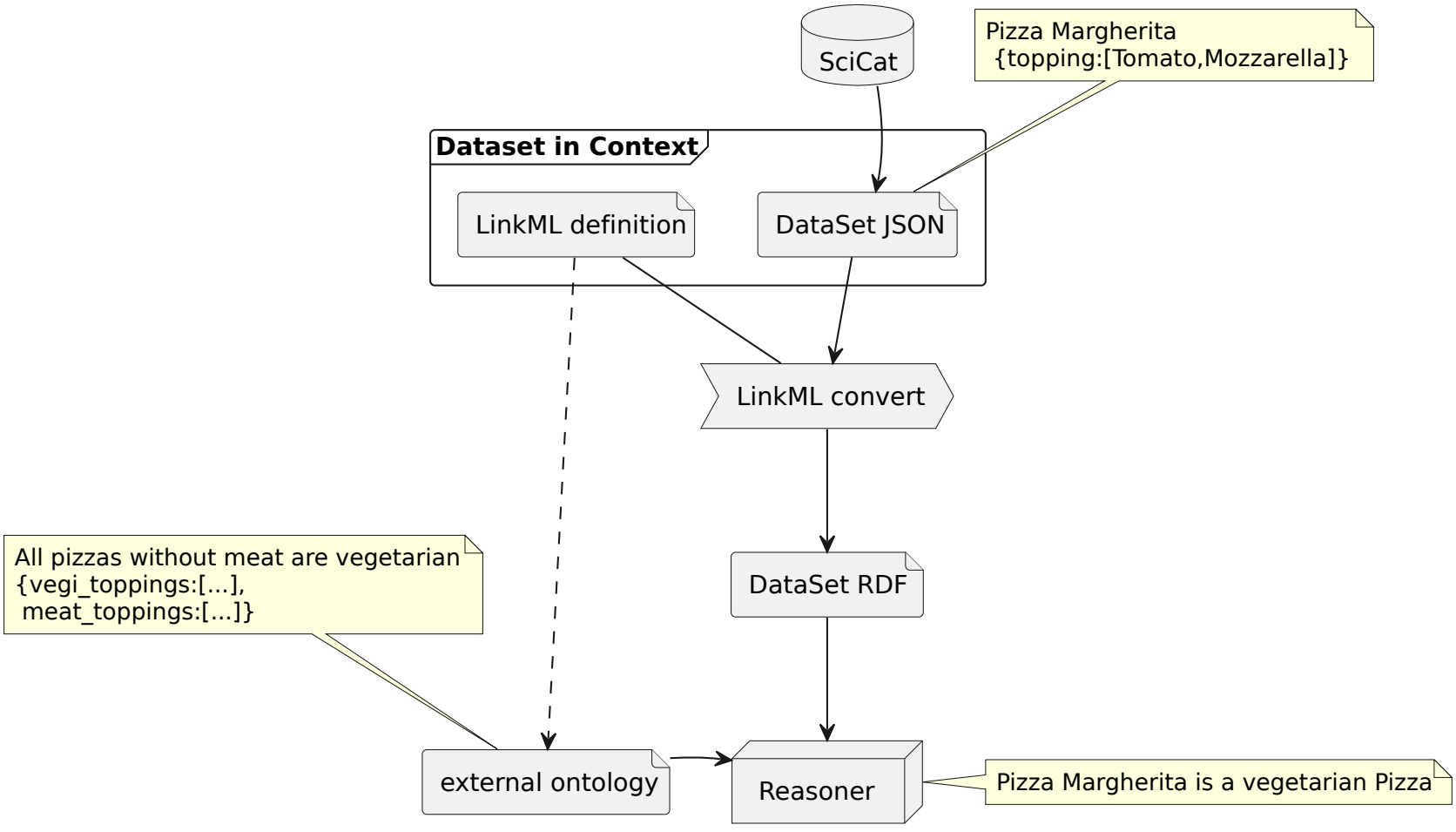
Dataset.yaml

```
id: https://desy.de/metadata/base_scicat
name: base_scicat_dataset
prefixes:
  schema: http://schema.org/
  linkml: https://w3id.org/linkml/
imports: linkml:types
default_range: string
classes:
  ScicatDataset:
    abstract: true
    attributes:
      owner:
        description: Owner or custodian of the dataset
        required: true
      ownerEmail:
        description: Email of the owner
      creationLocation:
        description: Unique location identifier
      dataFormat:
        description: Defines the data file format
      proposalId:
        description: The ID of the proposal
      sampleId:
         description: ID of the sample used when collecting the data
      instrumentId:
        description: ID of the instrument where the data was created
      ownerGroup:
        description: Defines the group which owns the data
      accessGroups:
        description: Defines the group which owns the data
      type:
        description: either 'raw' or 'derived'.
        range: type_options
        required: true
      scientificMetadata:
        range: object
enums:
  type_options:
    permissible_values:
      raw:
      derived:
```

PizzaDataset.yaml

```
id: https://desy.de/linkml/opendata/pizza2
name: pizza2
prefixes:
  linkml: https://w3id.org/linkml/

imports:
  - linkml:types
  - pizza
  - ../schema/base_scicat_dataset

classes:
  ScientificMetadataCommons:
    description: some optional common properties
    attributes:
        CustomParams:
          range: string
          required: false

  PizzaScientificMetadata:
    description: The metadata associated with Pizza
    is_a: Pizza
    mixins: [ScientificMetadataCommons]

  Pizzadataset:
    tree_root: true
    is_a: ScicatDataset
    description: The metadata associated with P65
    attributes:
      scientificMetadata:
        range: PizzaScientificMetadata
        required: true
```

Using DataSets as instances of an ontology: Pizza example



Loose Ends

- Where do we keep the information about the Schema?
- How to integrate with SciCat
  - imagine we wanted a LinkML validating endpoint
  - LinkML is a python codebase, not JavaScript. . .
- What are use-cases other institutes?
- Would it make sense to even think of an SPARQL endpoint for SciCat?

SPARQL query to find pizza with tomato topping

```
SELECT ?x
WHERE {
?x rdfs:subClassOf+ pizza:Pizza .
?x rdfs:subClassOf [
a owl:Restriction ;
owl:onProperty pizza:hasTopping;
owl:someValuesFrom pizza:TomatoTopping
]
}
```