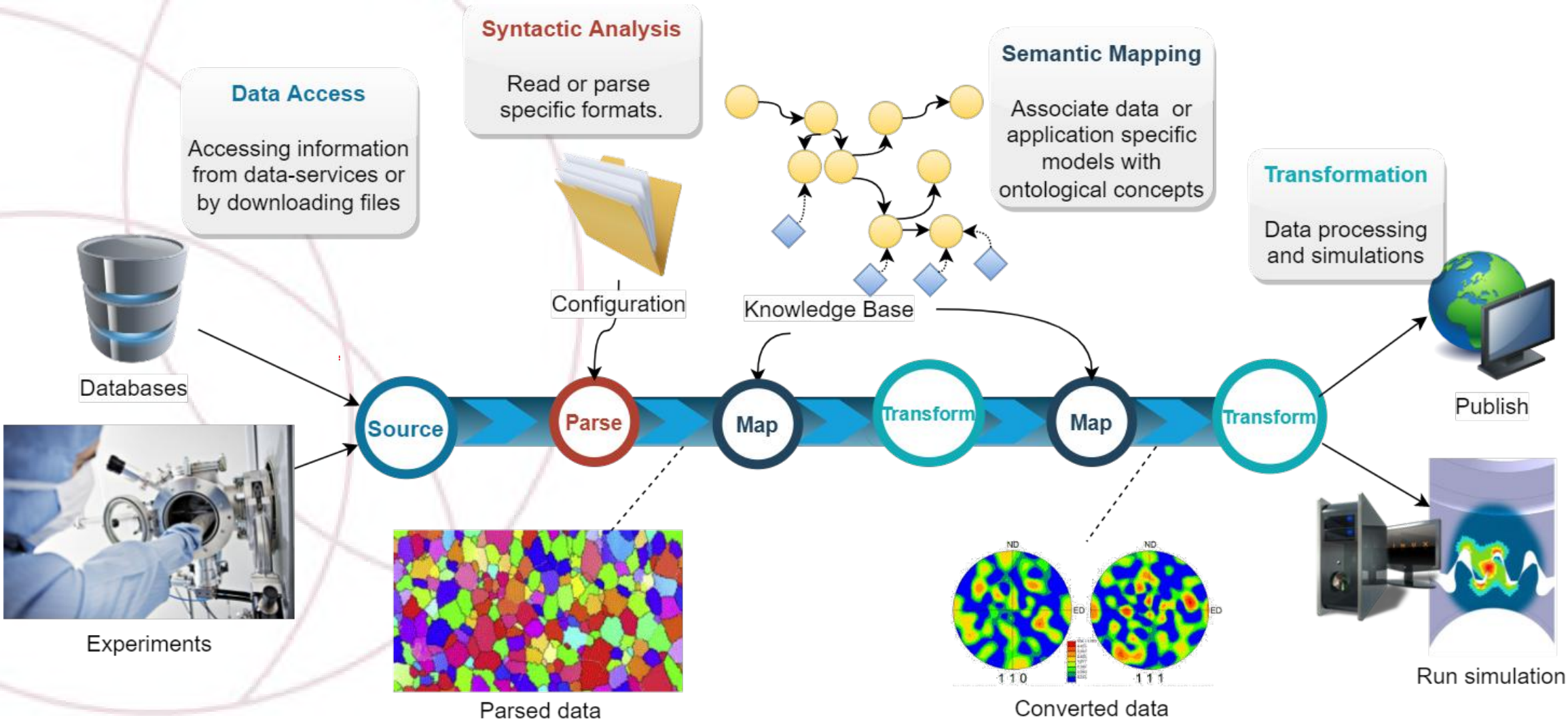
Connecting things OTEAPI

Thomas F. Hagelien (SINTEF)
Jesper Friis (SINTEF)

Connecting things

OTEAPI

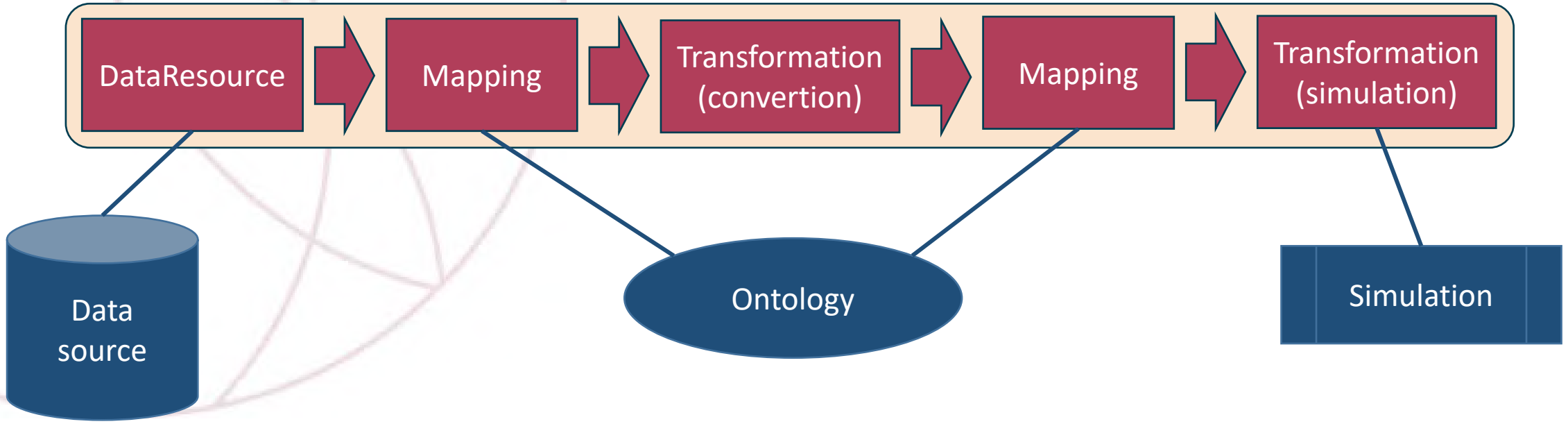
Data flow from source to receiver



OTEAPI

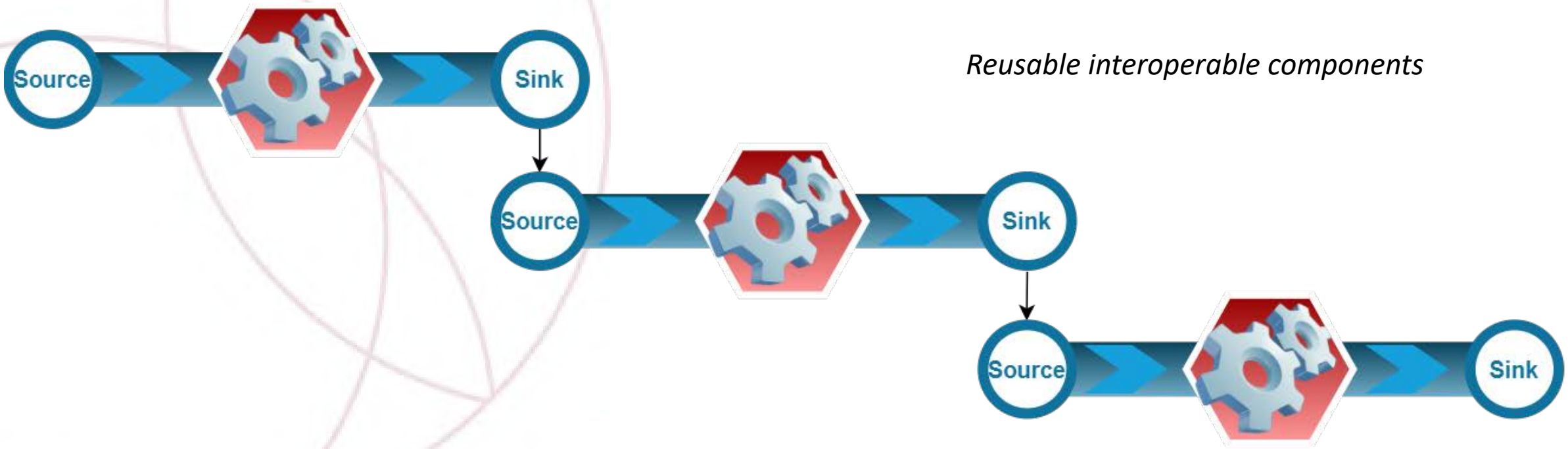
Pipeline

- Pipe and filter design pattern
- Standardise on interfaces
- Reusable "filters" that can be combined as needed (loosely coupled)
- Relies on (but is agnostic to) underlying interoperability framework
- Plugin-based (strategy design pattern)



Modelling workflows

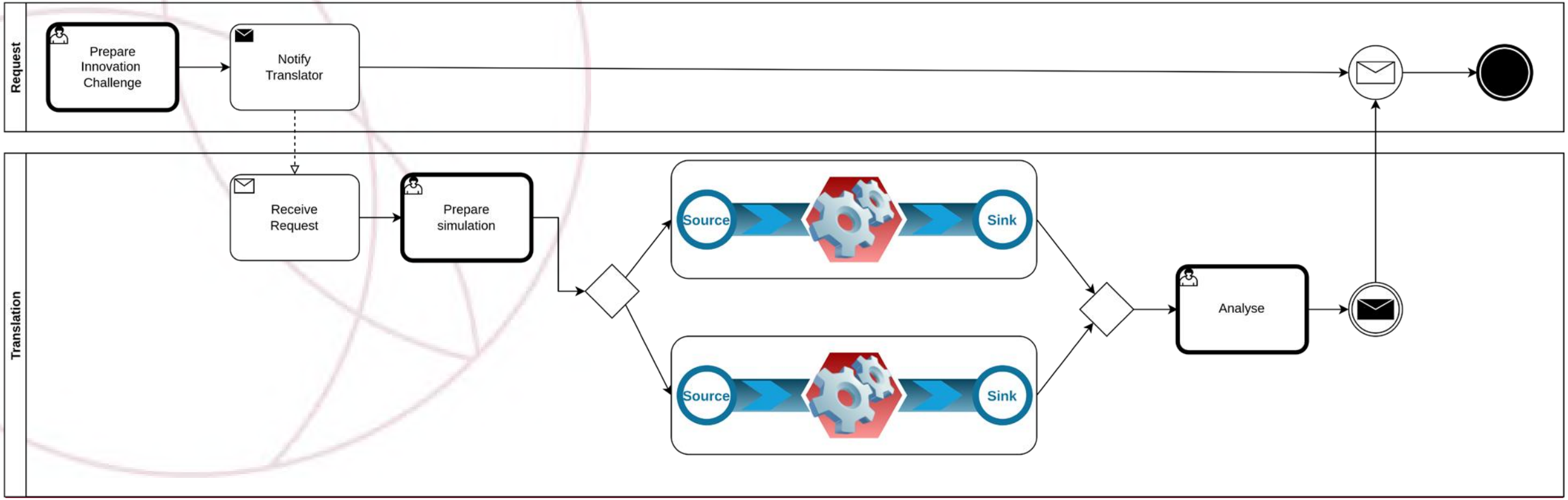
Pipelines can be mixed and combined into workflows



Reusable interoperable components

The bigger picture

Pipeline: data flow
Workflow: process flow



Types of "filters" (strategies)

Download
"downloadUrl"

Parse
"mediaType"

Resource
"accessUrl/accessService"

Connecting to data sources
- file-based / service-based
- follows the DCAT standard

Mapping
"mappingType"

Connecting to ontologies
- Key for semantic interoperability
- Depends on underlying intop system

Filter
"filterType"

Data selection

Function
"functionType"

Synchronious transformation.
Ex data conversion

Transformation
"transformationType"

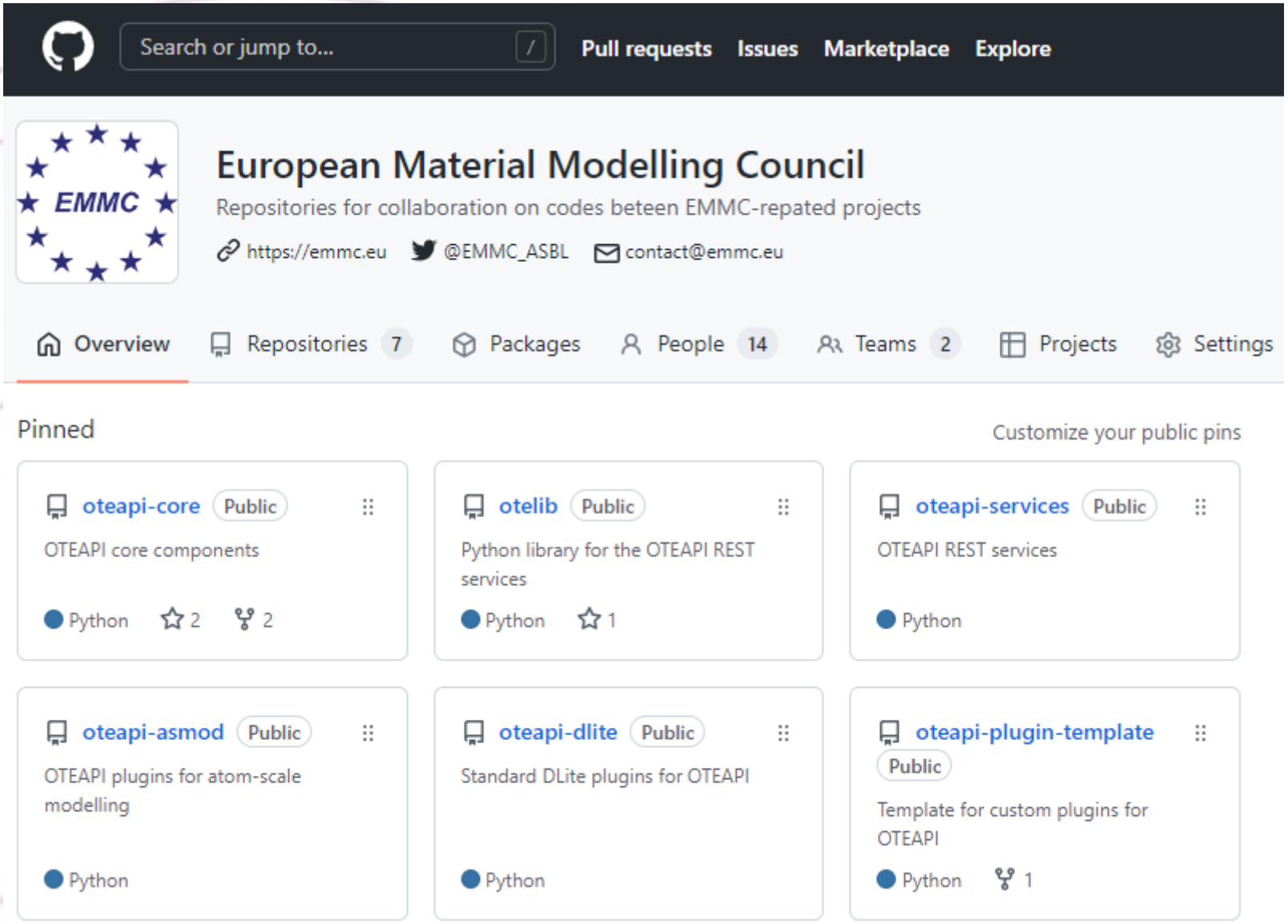
Asynchronious transformation.
Ex simulation

Examples

Download/parse
- JSON, YAML…
- Excel, csv, …
- app-specific…

Resource
- SQL
- OPTIMADE
- …

Strategies are configured
-> declarative programming



https://github.com/EMMC-ASBL/

Hosted on GitHub

- Facilitate reuse in other projects
- MIT license

A set of repositories

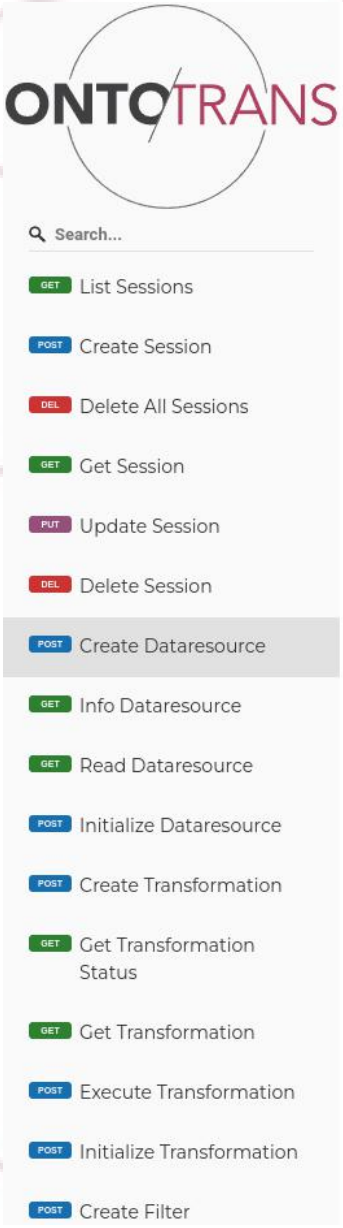
- oteapi-core
- oteapi-services (REST API)
- otelib (Python API)
- oteapi-plugin-template
- Plugins
  - oteapi-dlite
  - oteapi-asmod
  - …

oteapi-core

- A plugin system for loading the standard strategies, as well as third party strategies
- Data models for configuring the strategies
- A Python library, through which the data can be accessed
- A set of standard strategies

oteapi-services: REST API

Facilitate creation of pipelines



Create Dataresource

Register an external data resource.

An external data resource can be any data distribution provider that provides services of obtaining information through queries, REST APIs or other protocol, or directly downloadable artifacts (files) through data exchange procolols (such as sftp, https etc...)

If the resource URL is as direct link to a downloadable file, set the downloadURL property, otherwise set the accessURL the service and specify the service name with the mediaType property.

QUERY PARAMETERS

- `session_id` — string (Session Id)

REQUEST BODY SCHEMA: application/json

- `configuration` — object (Configuration)
  Default: `{}`
  Model-specific configuration options which can either be given as key/value-pairs or set as attributes.
- `description` — string (Description)
  Default: `"Resource Strategy Data Configuration.\n\n Important:\n Either of the pairs of attributes `downloadUrl`/`mediaType` or\n `accessUrl`/`accessService` MUST be specified.\n\n "`
  A description of the configuration model.
- `downloadUrl` — string <uri> (Downloadurl) `[ 1 .. 65536 ] characters`
  Definition: The URL of the downloadable file in a given format. E.g. CSV file or RDF file.
  Usage: `downloadURL` *SHOULD* be used for the URL at which this distribution is available directly, typically through a HTTPS GET request or SFTP.
- `mediaType` — string (Mediatype)
  The media type of the distribution as defined by IANA [IANA-MEDIA-TYPES].
  Usage: This property *SHOULD* be used when the media type of the distribution is defined in IANA [IANA-MEDIA-TYPES].
- `accessUrl` — string <uri> (Accessurl) `[ 1 .. 65536 ] characters`
  A URL of the resource that gives access to a distribution of the dataset. E.g. landing page, feed, SPARQL endpoint.

POST /api/v1/dataresource/

Request samples

Payload

Content type
application/json

Copy Expand all Collapse

```
{
  "configuration": { },
  "description": "Resource Strategy Data Configuration.\n\n
  "downloadUrl": "http://example.com",
  "mediaType": "string",
  "accessUrl": "http://example.com",
  "accessService": "string",
  "license": "string",
  "accessRights": "string",
  "publisher": "string"
}
```

Response samples

200 404 422

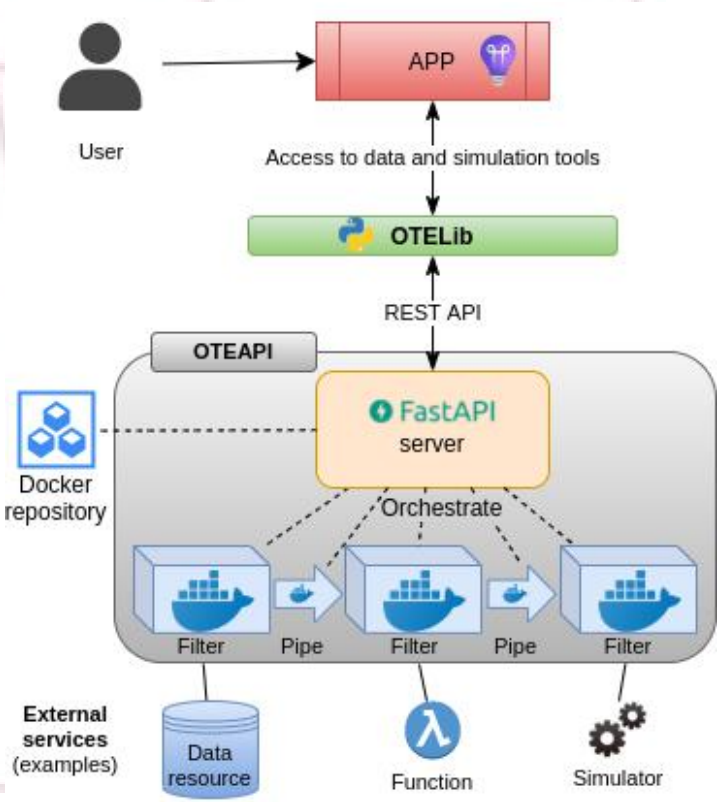
Content type
application/json

Copy Expand all Collapse

```
{
  "resource_id": "string"
}
```

OTELib

Python interface to OTEAPI for constructing executable pipelines




```python
from otelib import OTEClient

# To set up a pipeline, you will first have to
# connect to a running OTE server using the OTEClient
client = OTEClient('http://localhost:8080')

# Define data resource
data_resource = client.create_dataresource(
    downloadUrl="https://jpeg.org/images/jpegsystems-home.jpg",
    mediaType="image/jpeg",
)
# Define mappings
mapping = client.create_mapping(mappingType="mapping/demo", …)

# Define transformation step
transformation = client.create_transformation(transformation_type="script/dummy")


# Combine elements into a pipeline
pipeline = data_resource >> mapping >> transformation

# Execute pipeline
pipeline.get()
```

oteapi-plugin-template

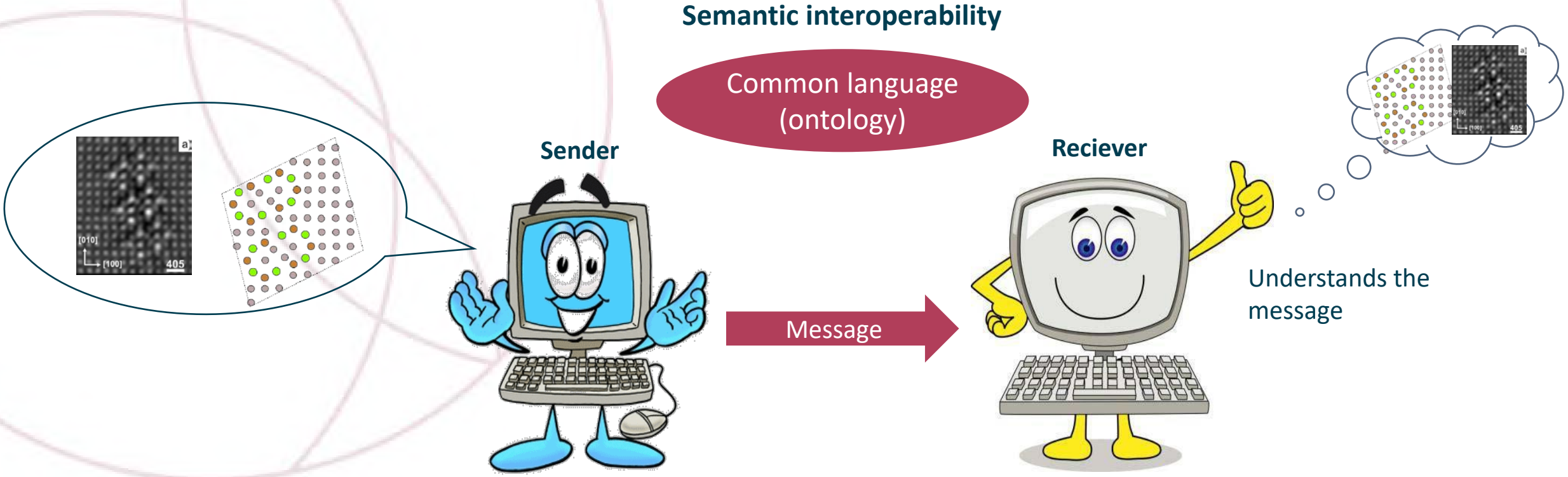
Simplify creation of plugins

Contains pre-configured features for:

- pip installable package
- Unit tests and Pre-commit configuration
- Continuous integration/continuous deployment
- Documentation via a MkDocs setup

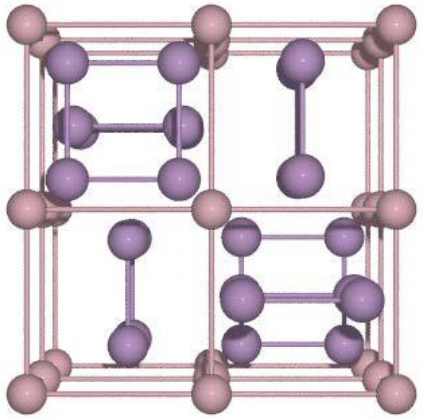
oteapi-dlite

Plugin using DLite as underlying interoperability framework



oteapi-asmod

Plugin for atomscale modelling



- Uses the Atomistic Simulation Environment (ASE) and DLite

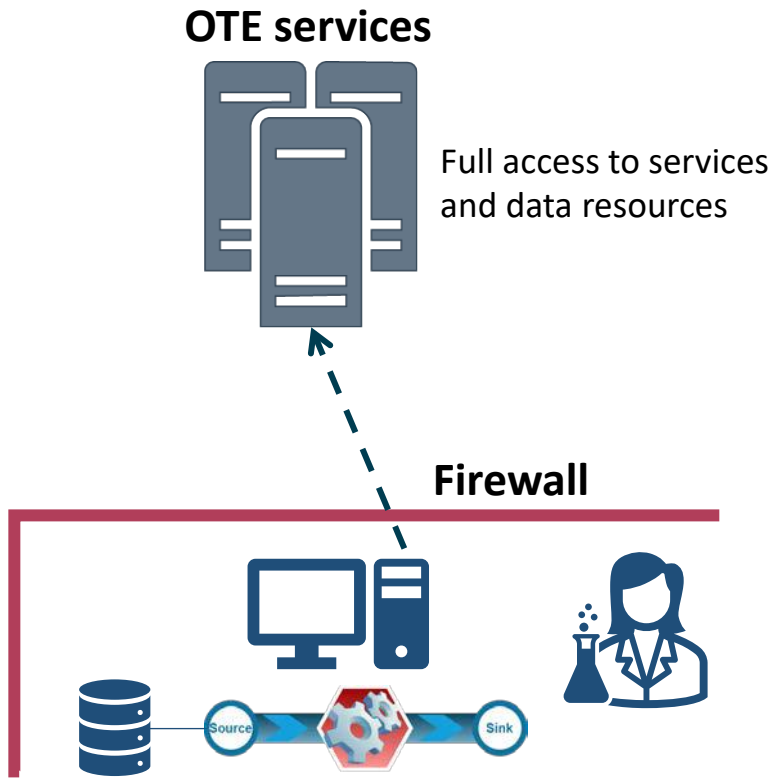
In addition private plugins are developed for the OntoTrans use cases

Under the hood of a pipeline

Data is actually not transferred between strategies…

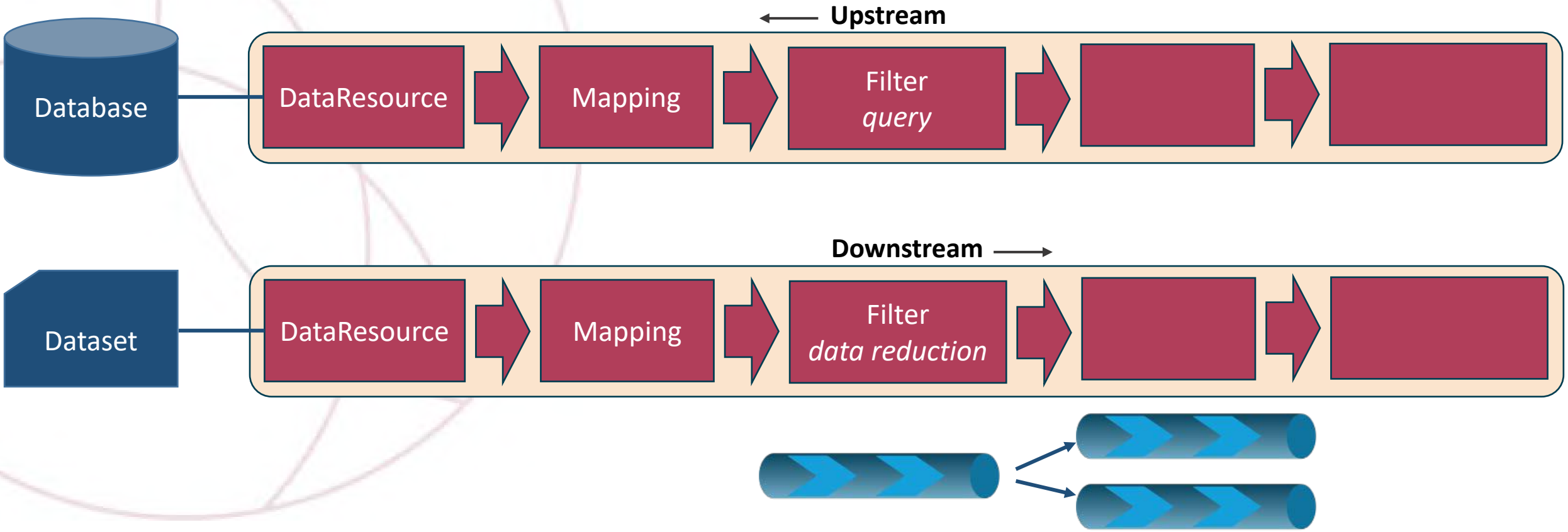
"A pipeline is a recipe for how data is handled"

- A data documentation system
- No data is transferred before pipeline execution
- Allow e.g. for execution behind a firewall



Filter strategy

Data reduction (both upstream and downstream)



Return to DLite…

Interoperability framework based on data models

An implementation of SOFT

Entity = data model

A data model that describes a self-contained unit.

An entity has:

- a unique identity
- a human description
- some dimensions
- some properties

What is an entity?

Something that exists by itself;
Something that is separate from other things.

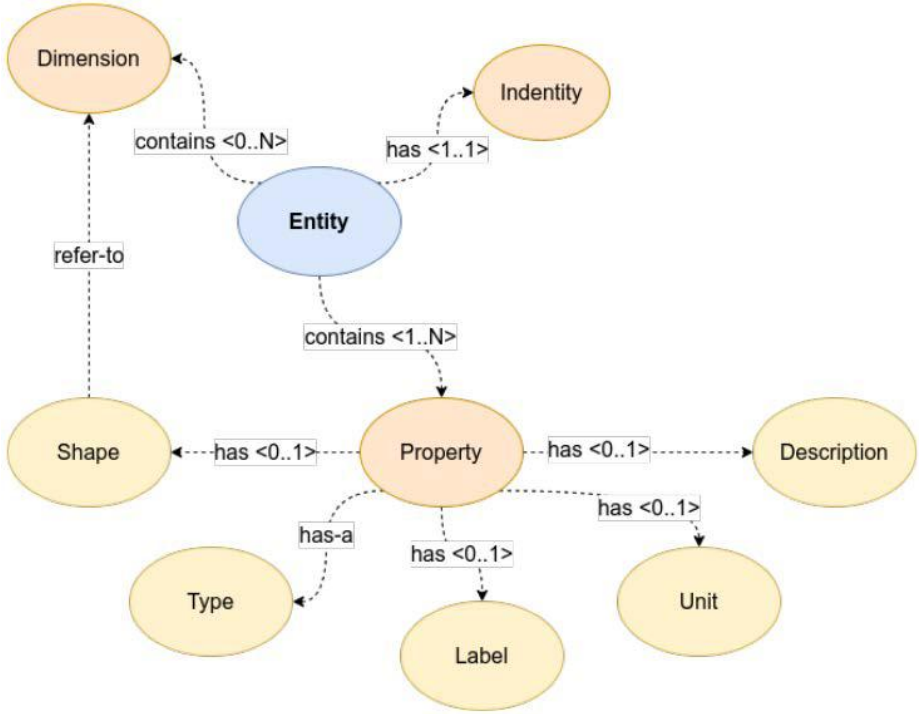
- Merriam-Webster

Schema for data models

Entity	
Identity (uri):	
Description:	

Dimensions	
Label (name)	**Description**

Properties				
Label (name)	**Type**	**Shape** (dims)	**Unit**	**Description**



Data model

Entity				
Identity (uri):	http://onto-ns.com/meta/0.1/Person			
Description:	A person.			
Dimensions				
Label (name)	**Description**			
N	Number of skills.			
Properties				
Label (name)	**Type**	**Shape (dims)**	**Unit**	**Description**
name	string			Full name.
age	float		year	Age or person.
skills	string	N		List of named skills.

Data model

Entity	
Identity (uri):	http://onto-ns.com/meta/0.1/Person
Description:	A person.

Dimensions	
Label (name)	**Description**
N	Number of skills.

Properties				
Label (name)	**Type**	**Shape (dims)**	**Unit**	**Description**
name	string			Full name.
age	float		year	Age or person.
skills	string	N		List of named s

Serialised as json

```
{
  "uri": "http://onto-ns.com/meta/0.1/Person",
  "description": "A person.",
  "dimensions": {
    "N": "Number of skills."
  },
  "properties": {
    "name": {
      "type": "string",
      "description": "Full name."
    },
    "age": {
      "type": "float",
      "unit": "year",
      "description": "Age of person."
    },
    "skills": {
      "type": "string",
      "shape": ["N"],
      "description": "List of skills."
    }
  }
}
```

Data model

Entity	
Identity (uri):	http://onto-ns.com/meta/0.1/Person
Description:	A person.

Dimensions	
Label (name)	**Description**
N	Number of skills.

Properties				
Label (name)	**Type**	**Shape (dims)**	**Unit**	**Description**
name	string			Full name.
age	float		year	Age or person.
skills	string	N		List of named skills.

Instance


```
{
  "88ca46ff-8404-48d9-b4b9-2140c6b3bdff": {
    "dimensions": {
      "N": 4
    },
    "properties": {
      "name": "Sherlock Homes",
      "age": 34.0,
      "skills": [
        "observing",
        "chemistry",
        "violin",
        "boxing"
      ]
    }
  }
}
```

Collection

Collection is an entity with

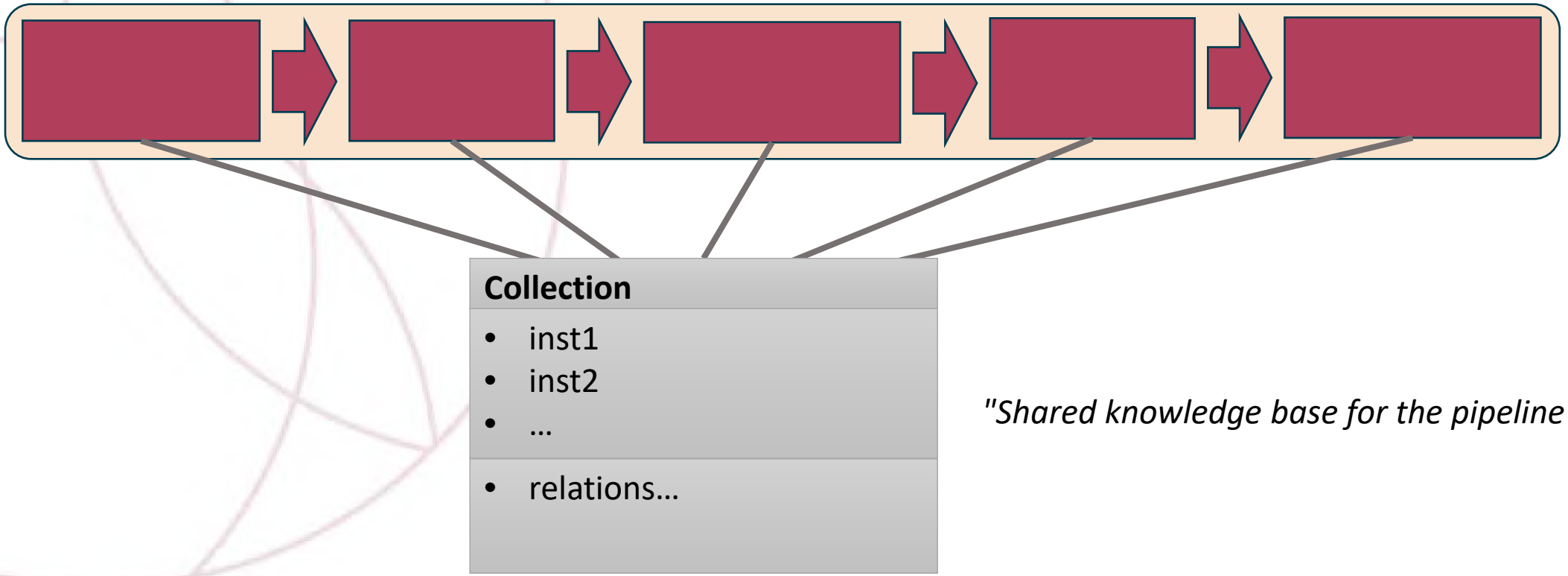
- references to instances
- relations between them

Can be used to build up arbitrary complex data structures

Collection		
Instances		
Label	Entity	UUID
SherlockHomes	http//…/Person	cad1d071-f3f8-44bf-b6f2-45b686dfd930
DrWatson	http//…/Person	d70697fb-e6de-4c85-b3d3-a5316f328234
Crime	http//…/Event	573adf17-02c3-4877-aafb-e7f1e12d239b
Moriarty	http//…/Person	9f97a7bc-a8dd-449a-999d-464d868fd013
Relations		
Subject	Predicate	Object
SherlockHomes	solves	Crime
DrWatson	helps	SherlockHomes
Moriarty	perform	Crime
…		

Using DLite with OTEAPI

A collection shared between all strategies

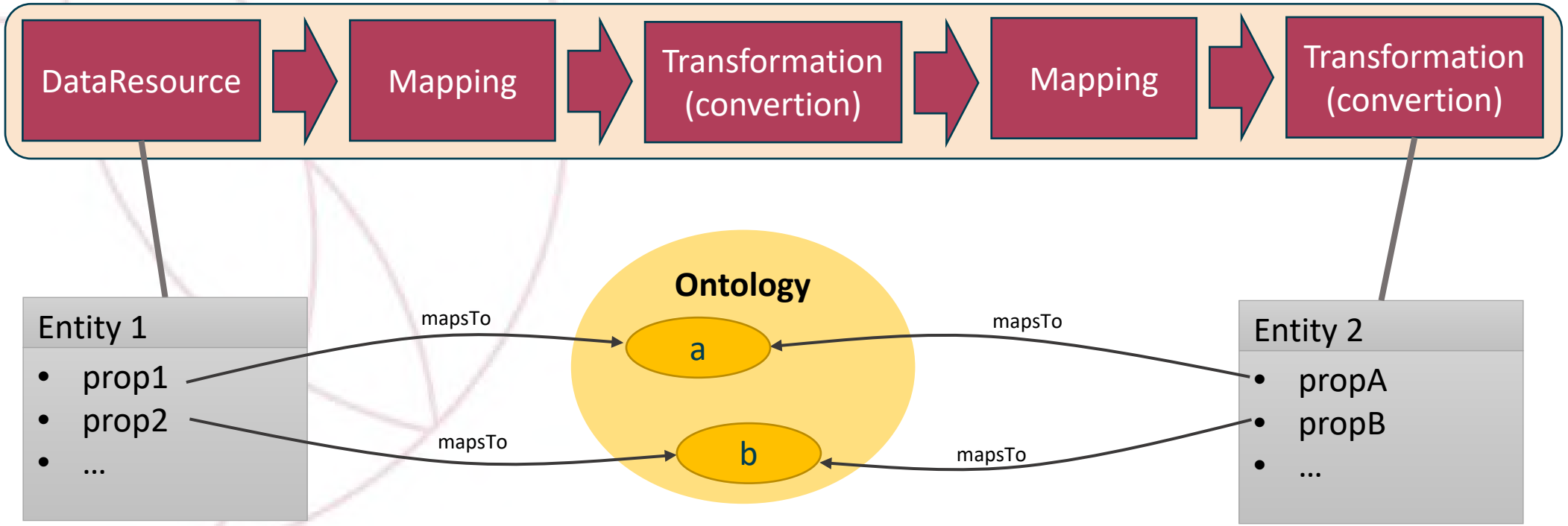


"Shared knowledge base for the pipeline"

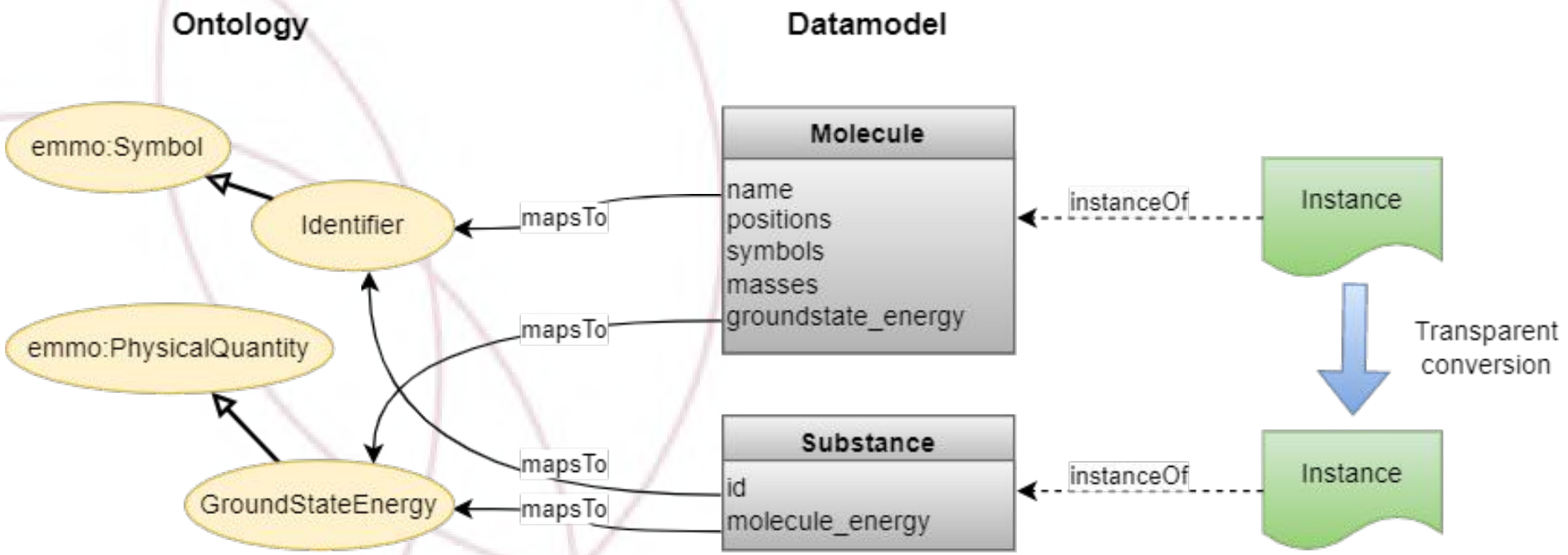
The RDF standard is used to express mappings

Interoperability

Conversion between data models – semantic interoperability



Example



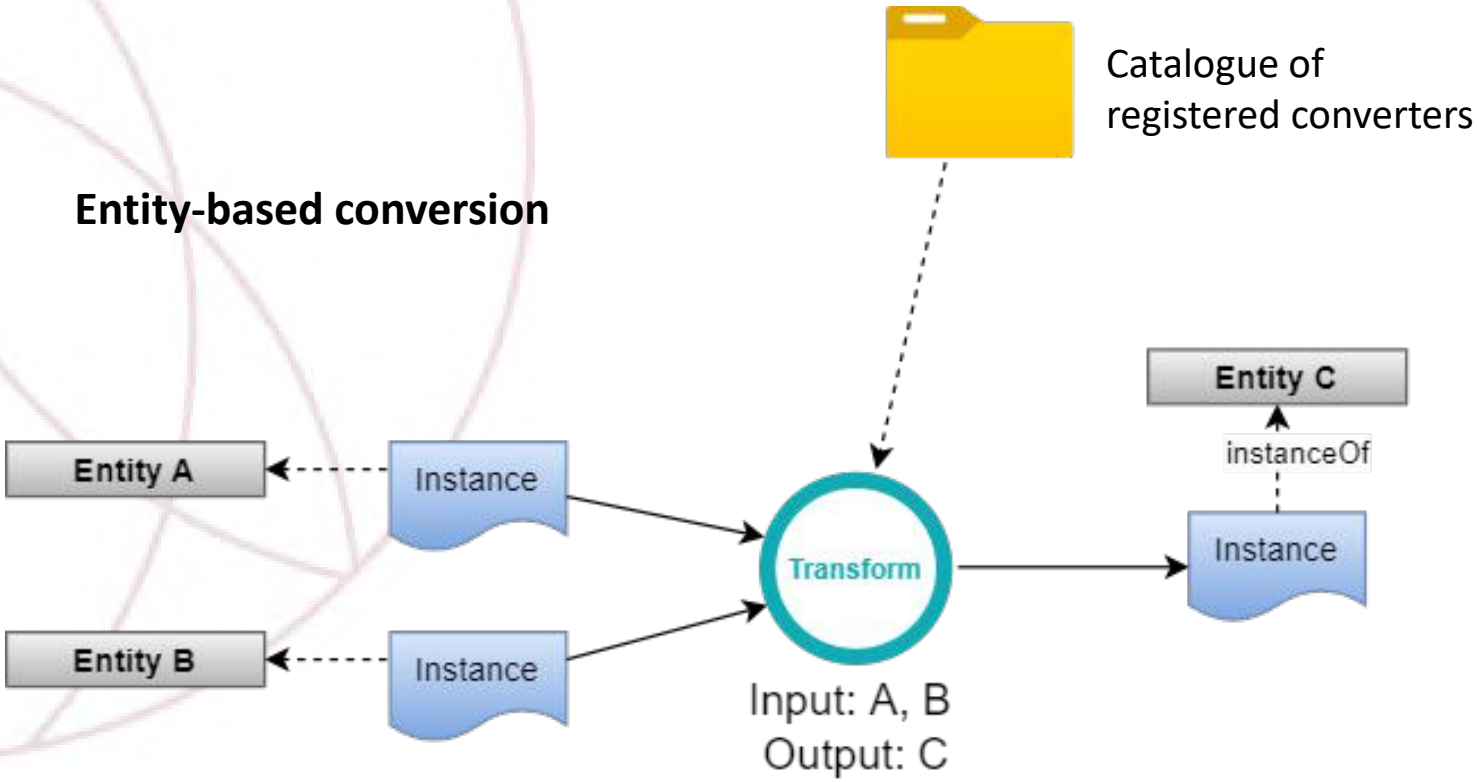
Some data

Provided as an instance of Molecule

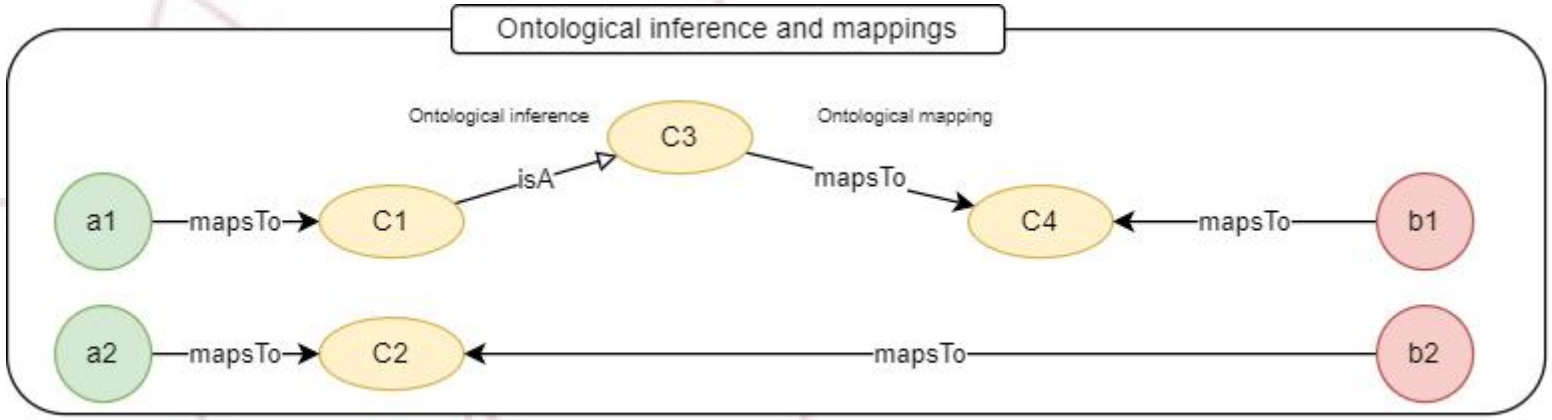
Wanted as an instance of Substance

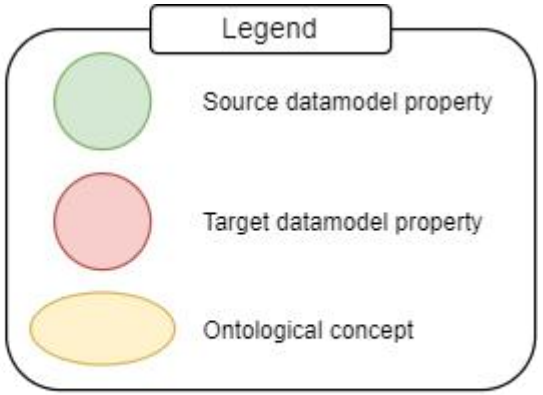
Entity-based conversions

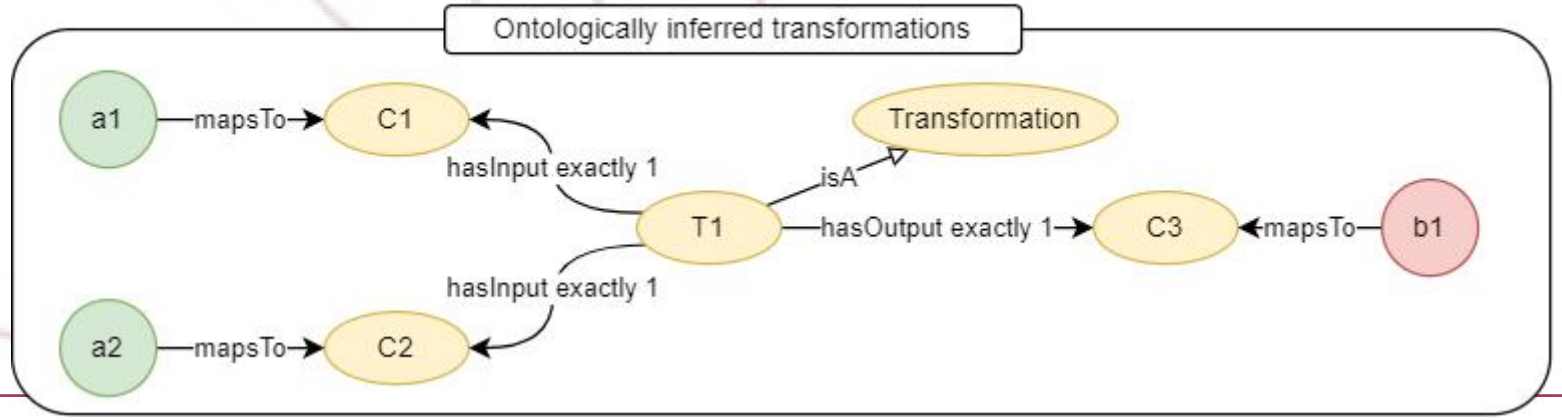
Supported by DLite – for complex cases



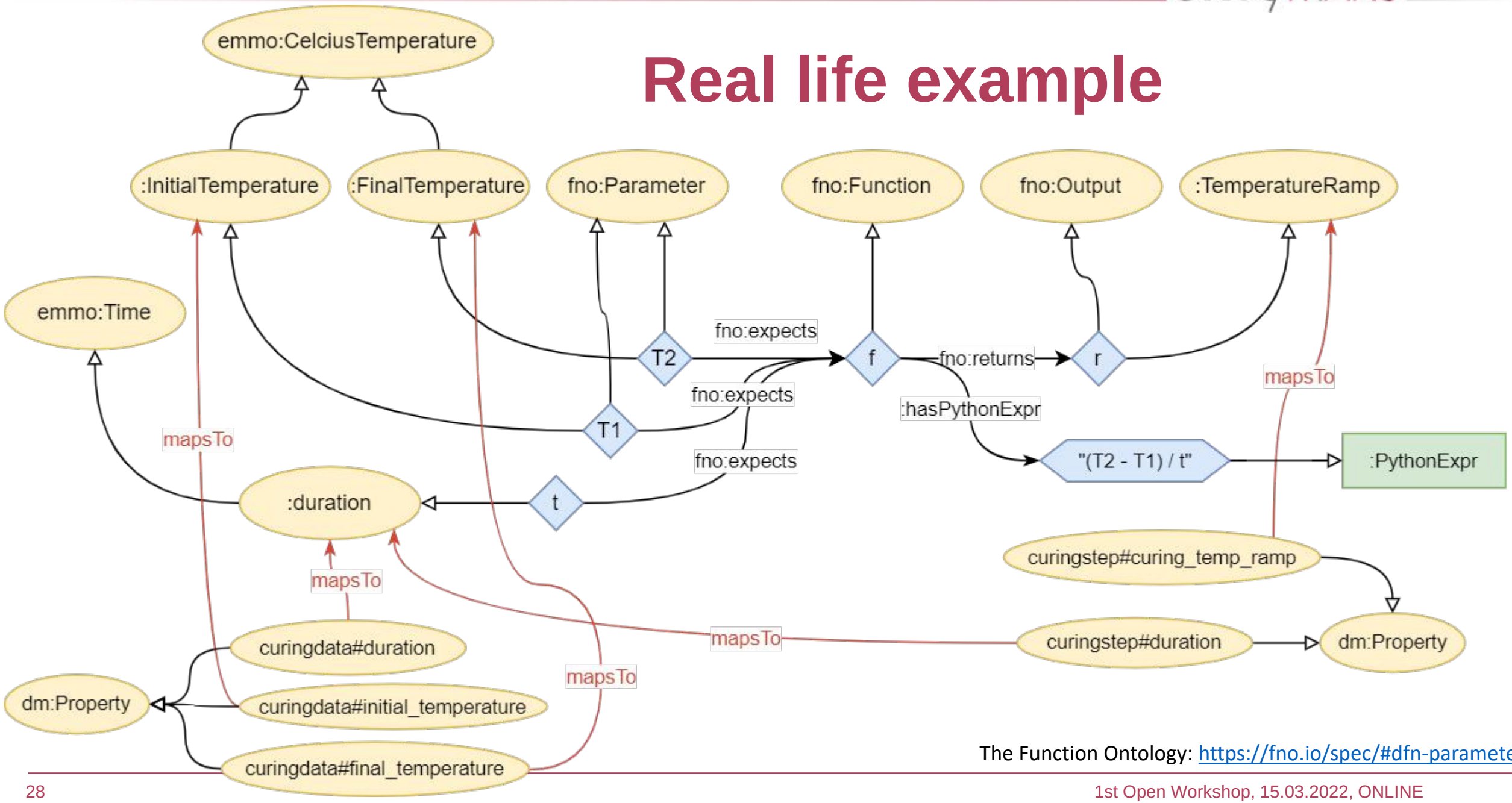
The really cool stuff…







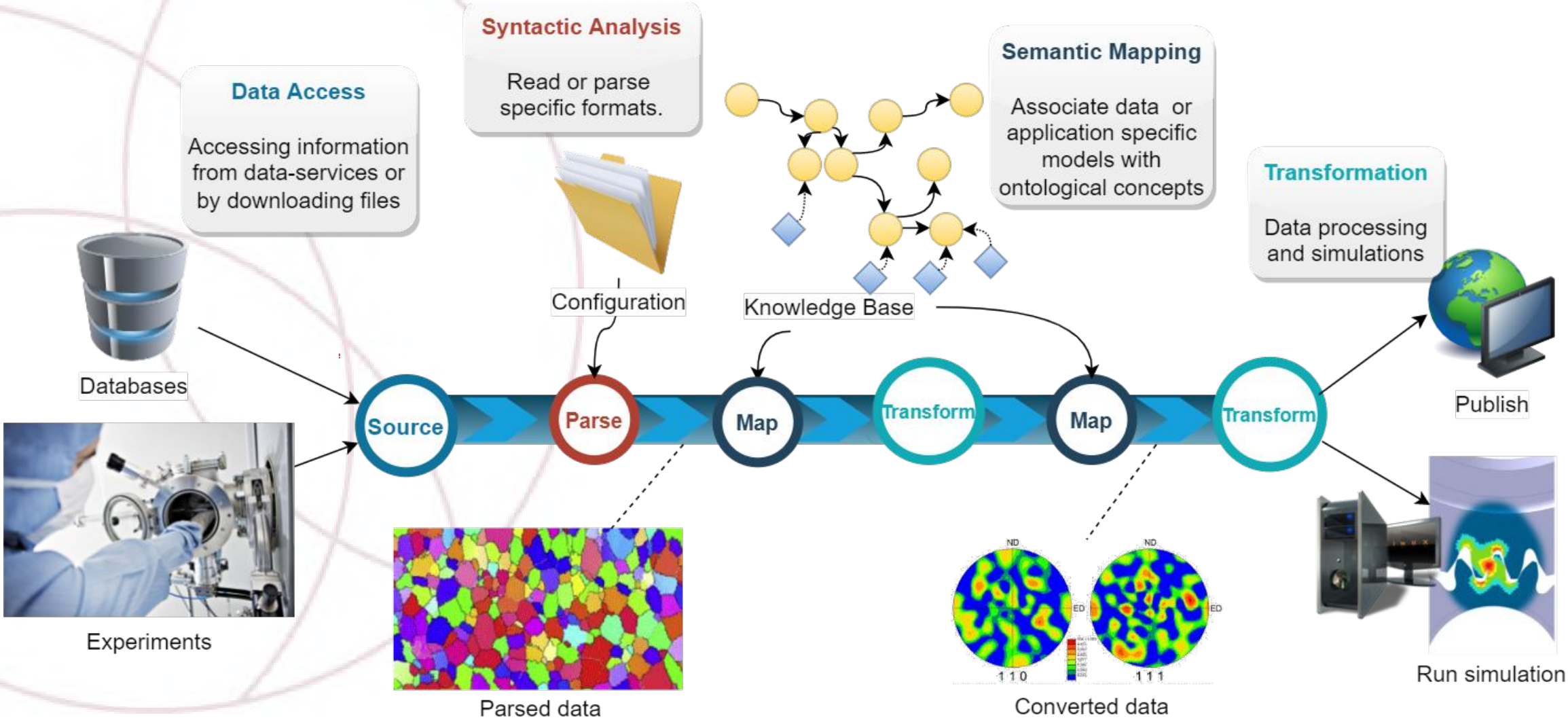
Real life example



The Function Ontology: https://fno.io/spec/#dfn-parameter

Summary: OTEAPI is framework for connecting things

Data flow from source to reciever



Thank you!

The OntoTrans project has received funding from the European Union‘s Horizon 2020 research and innovation programme under Grant Agreement No 862136.