IEEE TASK FORCE ON PROCESS MINING

# A Model-Driven Engineering perspective for the OCED metamodel

Daniel Calegari . Andrea Delgado
Instituto de Computación . Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

FACULTAD DE INGENIERÍA

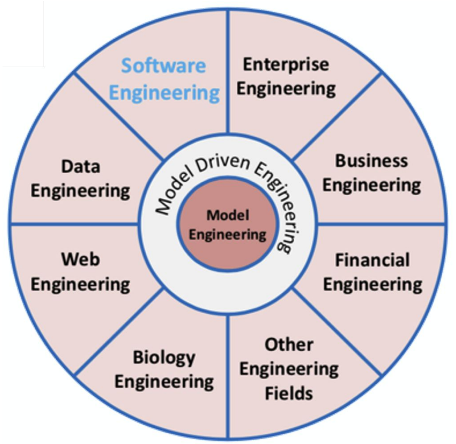
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

## Motivation

Model-Driven Engineering (**MDE**) emphasizes the specification of models conforming to metamodels and the use of transformations between them for various objectives, e.g., model refinement and code generation.

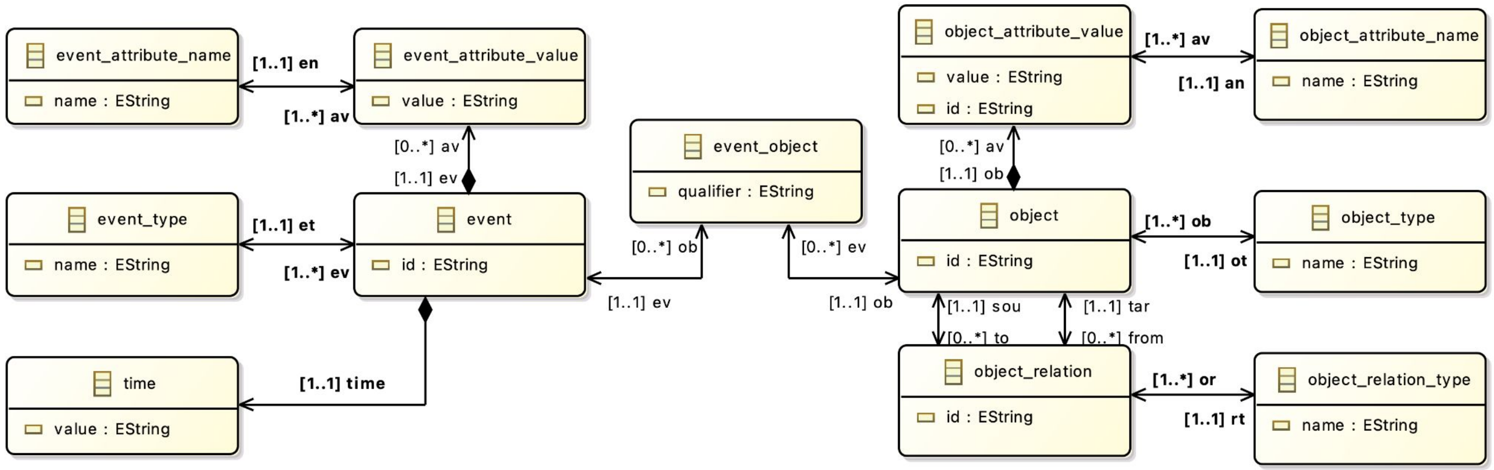
It provides a mature set of technical concepts and implemented technologies.

Adopting an **MDE perspective for the OCED proposal** could provide **several benefits**.



# An MDE perspective

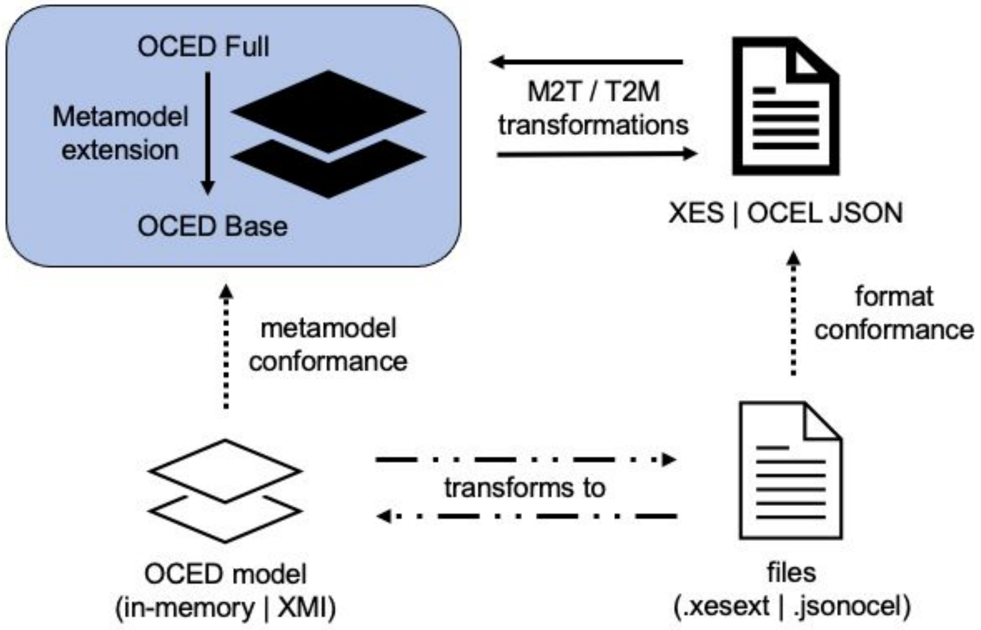
Ecore-based **reference implementation of OCED**



D. Calegari and A. Delgado: A Model-Driven Engineering perspective for the Object-Centric Event Data (OCED) metamodel. Objects Workshop @BPM`23

# An MDE perspective

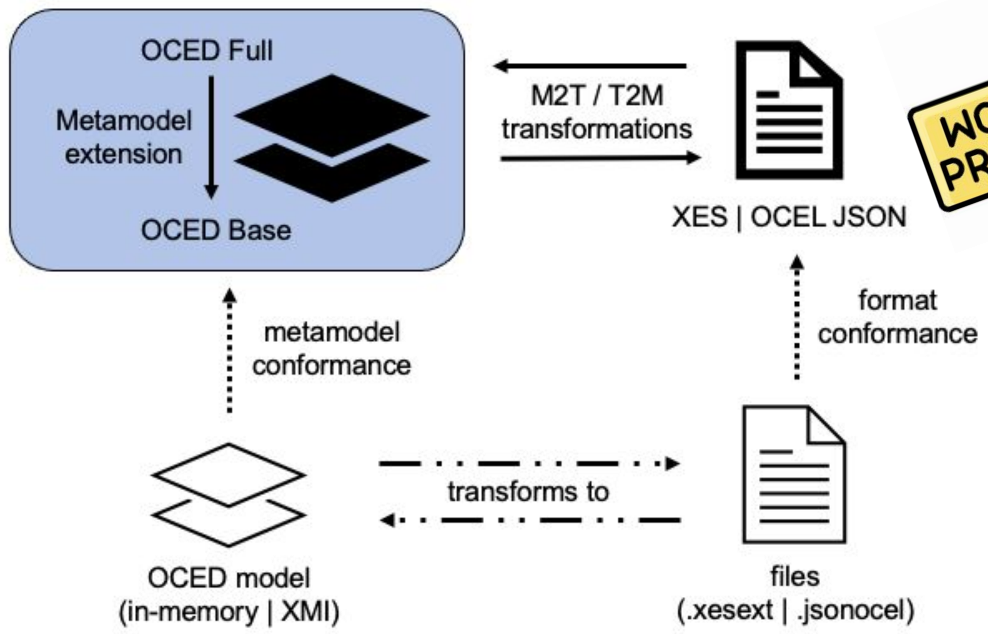
Java | Python



`https://gitlab.fing.edu.uy/open-coal/oced`

# An MDE perspective

Java | Python



**Other formats**
**-> OCEL 2**
**-> OCED-PG**

**Java/Python libraries**

`https://gitlab.fing.edu.uy/open-coal/oced`

## An MDE perspective

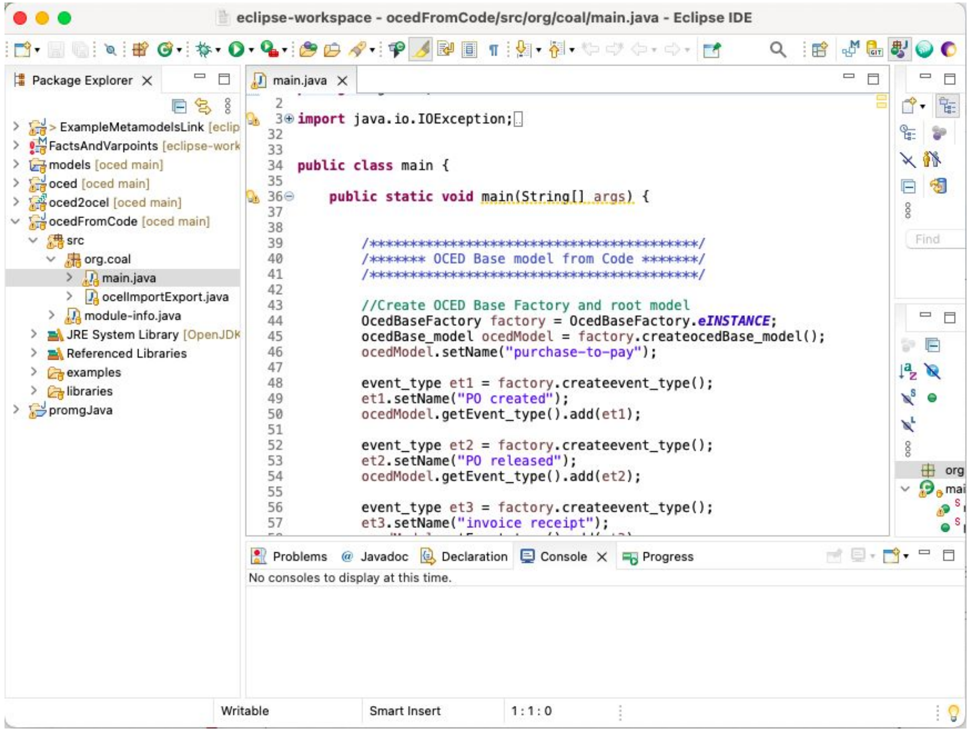
A **standard XMI** (*) representation

```
<?xml version="1.0" encoding="ASCII"?>

<ocedBase:ocedBase_model>

     <event id="99825">

          <event_object object="//@object.0" qualifier="CREATE"/>

     </event>

     <object id="4289" ... >

     </object>

</ocedBase:ocedBase_model>
```

**(*) XML Metadata Interchange (XMI) is an Object Management Group (OMG) standard for exchanging metadata information via XML**

# An MDE perspective

**Java** | Python



eclipse-workspace - ocedFromCode/src/org/coal/main.java - Eclipse IDE

Package Explorer

- ExampleMetamodelsLink [eclip
- FactsAndVarpoints [eclipse-work
- models [oced main]
- oced [oced main]
- oced2ocel [oced main]
- ocedFromCode [oced main]
  - src
    - org.coal
      - main.java
      - ocelImportExport.java
    - module-info.java
  - JRE System Library [OpenJDK
  - Referenced Libraries
  - examples
  - libraries
- promgJava

main.java

```java
import java.io.IOException;

public class main {

    public static void main(String[] args) {


        /*********************************************/
        /******* OCED Base model from Code *******/
        /*********************************************/

        //Create OCED Base Factory and root model
        OcedBaseFactory factory = OcedBaseFactory.eINSTANCE;
        ocedBase_model ocedModel = factory.createocedBase_model();
        ocedModel.setName("purchase-to-pay");

        event_type et1 = factory.createevent_type();
        et1.setName("PO created");
        ocedModel.getEvent_type().add(et1);

        event_type et2 = factory.createevent_type();
        et2.setName("PO released");
        ocedModel.getEvent_type().add(et2);

        event_type et3 = factory.createevent_type();
        et3.setName("invoice receipt");
```

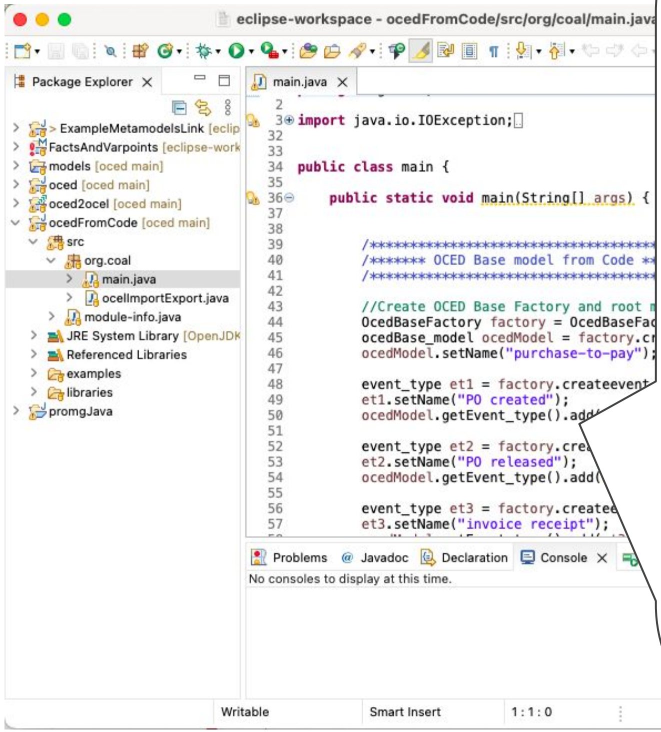
Problems | Javadoc | Declaration | Console | Progress

```
No consoles to display at this time.
```

Writable | Smart Insert | 1 : 1 : 0

# An MDE perspective

**Java** | Python



eclipse-workspace - ocedFromCode/src/org/coal/main.java

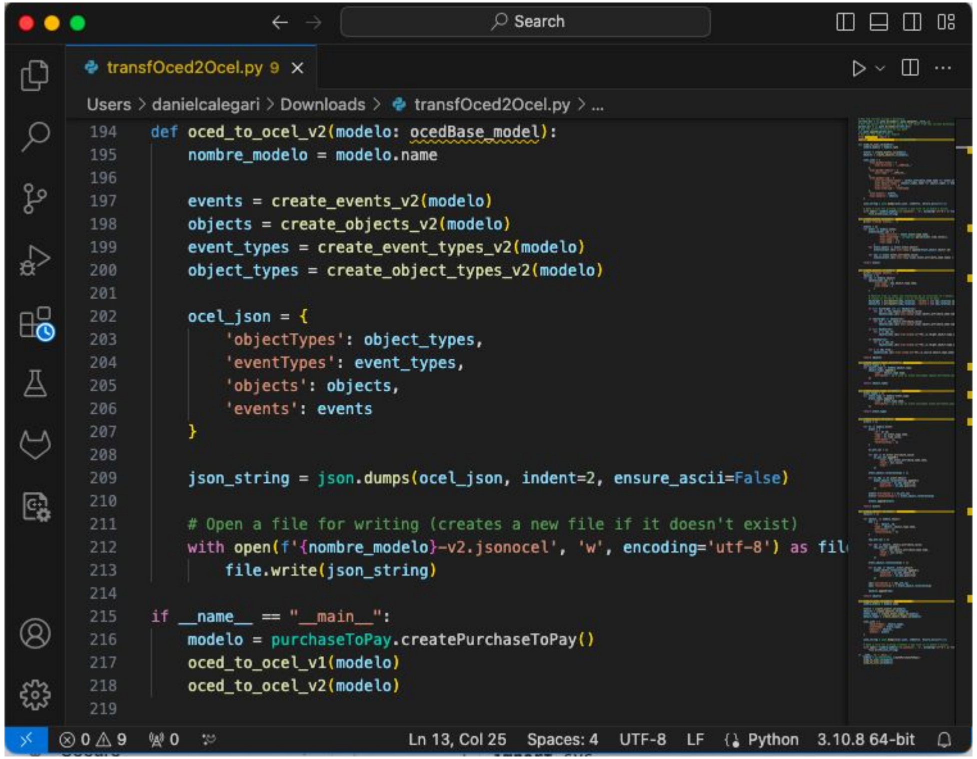
```
import java.io.IOException;


public class main {

    public static void main(String[] args) {


        /************************************
        /******* OCED Base model from Code **
        /************************************

        //Create OCED Base Factory and root m
        OcedBaseFactory factory = OcedBaseFac
        ocedBase_model ocedModel = factory.cr
        ocedModel.setName("purchase-to-pay");

        event_type et1 = factory.createevent
        et1.setName("PO created");
        ocedModel.getEvent_type().add(

        event_type et2 = factory.cre
        et2.setName("PO released");
        ocedModel.getEvent_type().add(

        event_type et3 = factory.createe
        et3.setName("invoice receipt");
```

No consoles to display at this time.

```
event ev1 = createevent();

ev1.setId("99825");

object ob1 = createobject();

ob1.setId("4289");

event_object eo1 = createev_obj();

eo1.setQualifier("CREATE");

eo1.setEvent(ev1);

eo1.setObject(ob1);

...

ocedBase_model          m          =
import_baselog("...");

export_log(m,"xxx",fileFormat.JSON);
```

# An MDE perspective

Java | **Python**

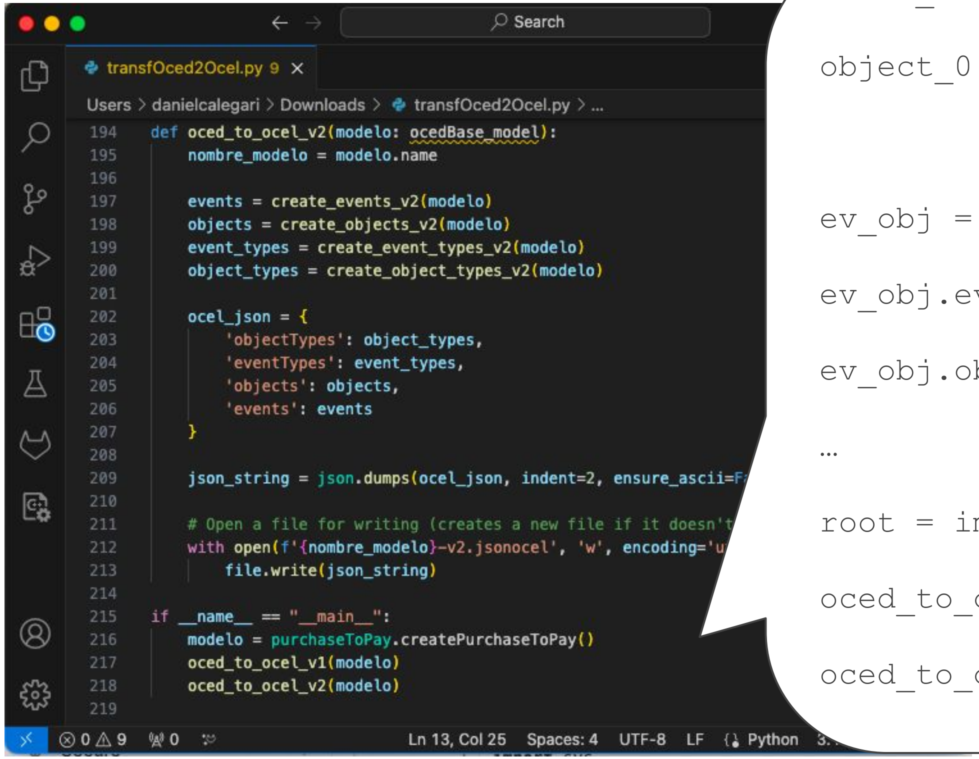


```
def oced_to_ocel_v2(modelo: ocedBase_model):
    nombre_modelo = modelo.name

    events = create_events_v2(modelo)
    objects = create_objects_v2(modelo)
    event_types = create_event_types_v2(modelo)
    object_types = create_object_types_v2(modelo)

    ocel_json = {
        'objectTypes': object_types,
        'eventTypes': event_types,
        'objects': objects,
        'events': events
    }

    json_string = json.dumps(ocel_json, indent=2, ensure_ascii=False)

    # Open a file for writing (creates a new file if it doesn't exist)
    with open(f'{nombre_modelo}-v2.jsonocel', 'w', encoding='utf-8') as fil
        file.write(json_string)

if __name__ == "__main__":
    modelo = purchaseToPay.createPurchaseToPay()
    oced_to_ocel_v1(modelo)
    oced_to_ocel_v2(modelo)
```

Search
transfOced2Ocel.py 9
Users > danielcalegari > Downloads > transfOced2Ocel.py > ...
0 ⚠ 9 0
Ln 13, Col 25 Spaces: 4 UTF-8 LF Python 3.10.8 64-bit

# An MDE perspective

Java | **Python**



transfOced2Ocel.py 9

Users > danielcalegari > Downloads > transfOced2Ocel.py > ...

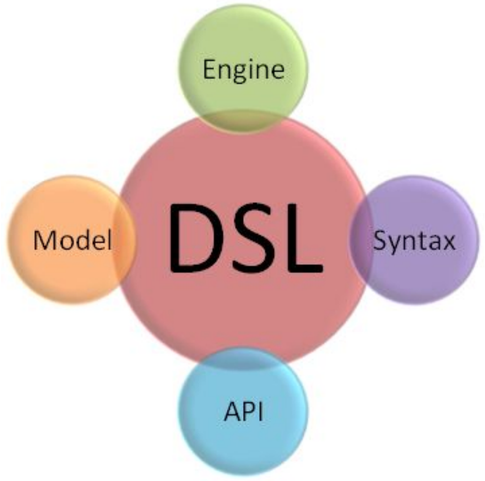
```python
def oced_to_ocel_v2(modelo: ocedBase_model):
    nombre_modelo = modelo.name

    events = create_events_v2(modelo)
    objects = create_objects_v2(modelo)
    event_types = create_event_types_v2(modelo)
    object_types = create_object_types_v2(modelo)

    ocel_json = {
        'objectTypes': object_types,
        'eventTypes': event_types,
        'objects': objects,
        'events': events
    }

    json_string = json.dumps(ocel_json, indent=2, ensure_ascii=F

    # Open a file for writing (creates a new file if it doesn't
    with open(f'{nombre_modelo}-v2.jsonocel', 'w', encoding='u
        file.write(json_string)

if __name__ == "__main__":
    modelo = purchaseToPay.createPurchaseToPay()
    oced_to_ocel_v1(modelo)
    oced_to_ocel_v2(modelo)
```

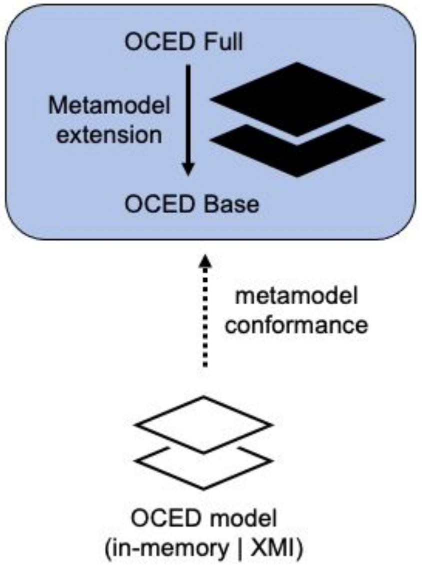
Ln 13, Col 25   Spaces: 4   UTF-8   LF   {} Python

```
event_0 = event(id='99825')

object_0 = object(id='4289')


ev_obj = event_object("CREATE")

ev_obj.event = event_0

ev_obj.object = object_0

…

root = import_XMI(path)

oced_to_ocel_v1(root)

oced_to_ocel_v2(root)
```

## Some reflections

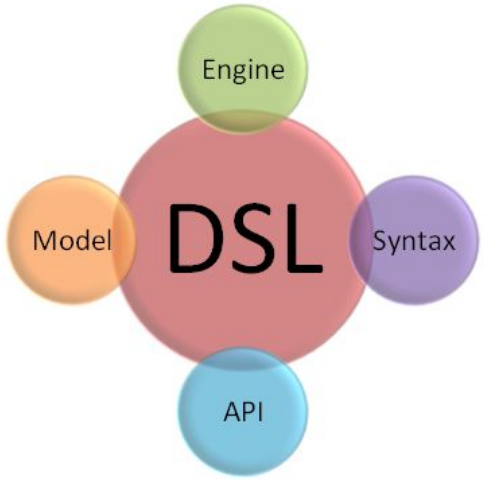
domain-specific



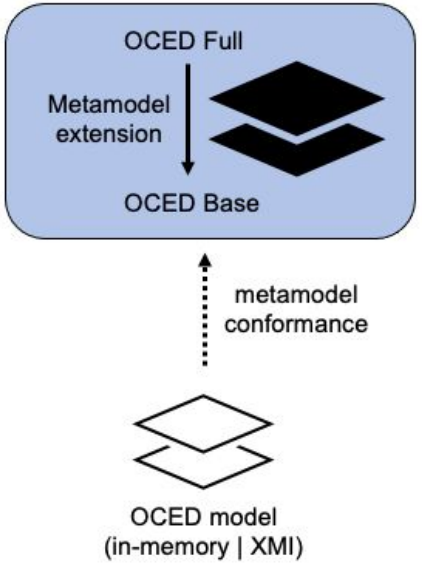


# Some reflections

domain-specific

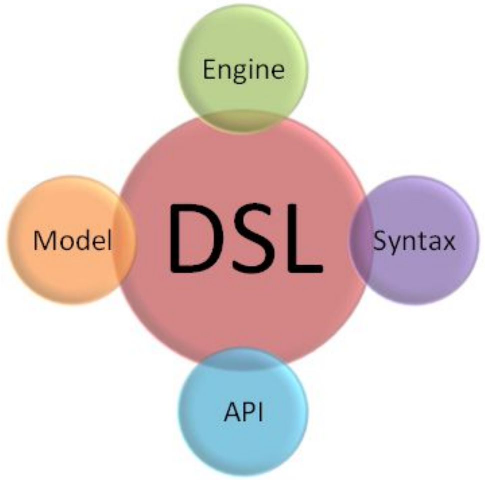


domain-**agnostic**?



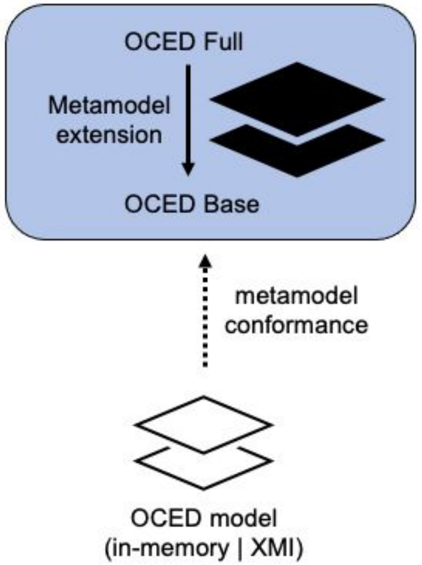
## Some reflections

domain-specific



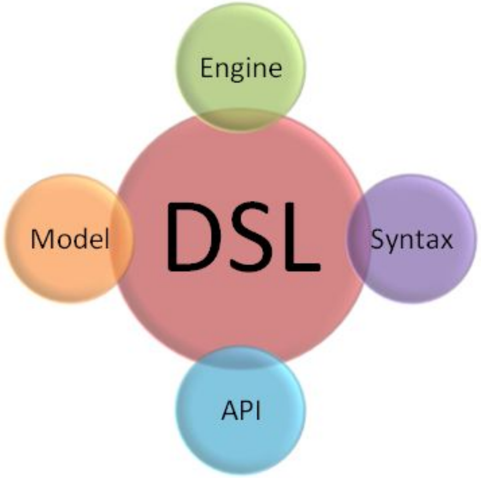
how to **map**?

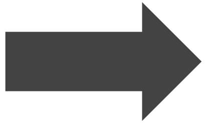
domain-**agnostic**?



## Some reflections

domain-specific

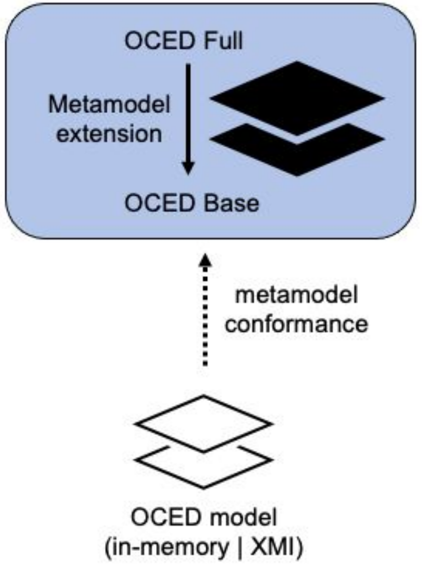




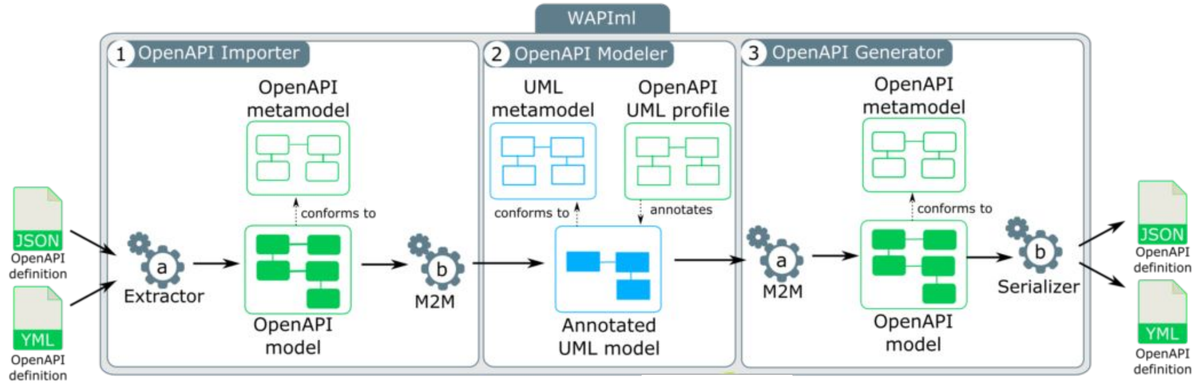
how to **map**?

how to **extend**?
(e.g., relations as first-class citizens)

domain-**agnostic**?

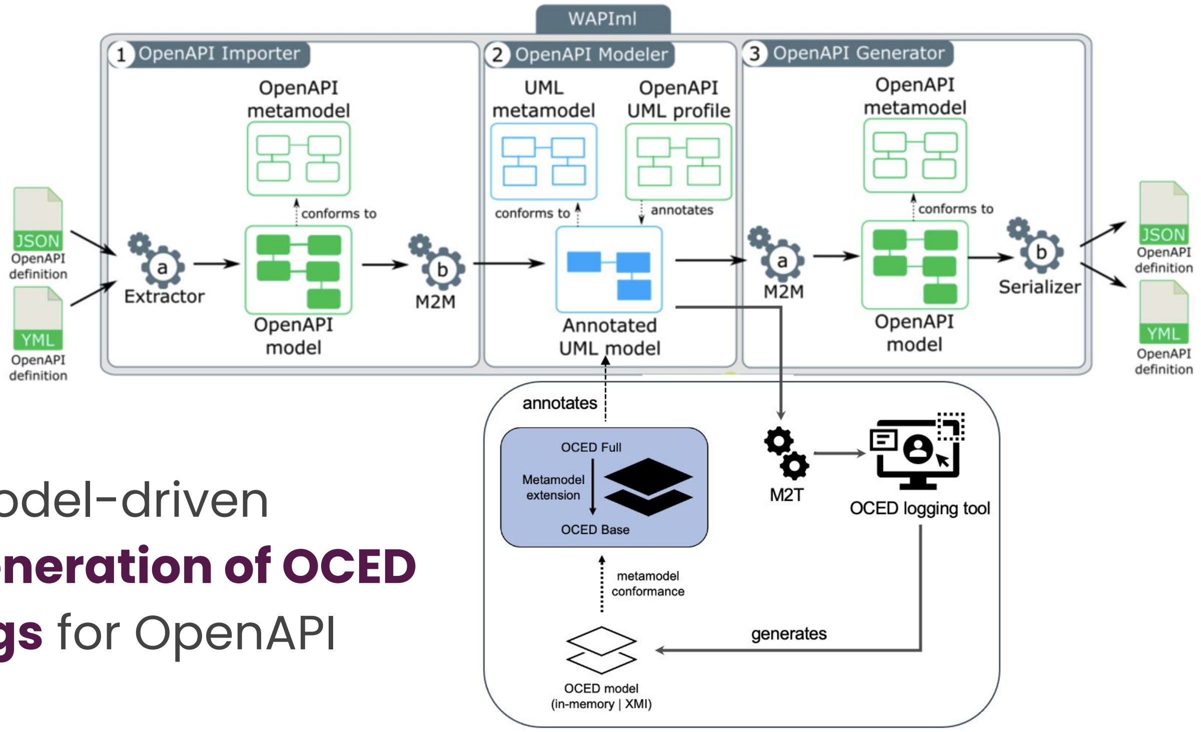


## Let's give an example...



H. Ed-douibi, J. L. Cánovas, F. Bordeleau and J. Cabot, "WAPIml: Towards a Modeling Infrastructure for Web APIs," *2019 ACM/IEEE 22nd Intl. Conf. MODELS*, 2019, pp. 748-752

# Let's give an example...



model-driven **generation of OCED logs** for OpenAPI

# Thanks!

Daniel Calegari . Andrea Delgado
Inco . Fing . Udelar
coal@fing.edu.uy
www.fing.edu.uy/inco/grupos/coal

FACULTAD DE INGENIERÍA

UNIVERSIDAD DE LA REPÚBLICA URUGUAY