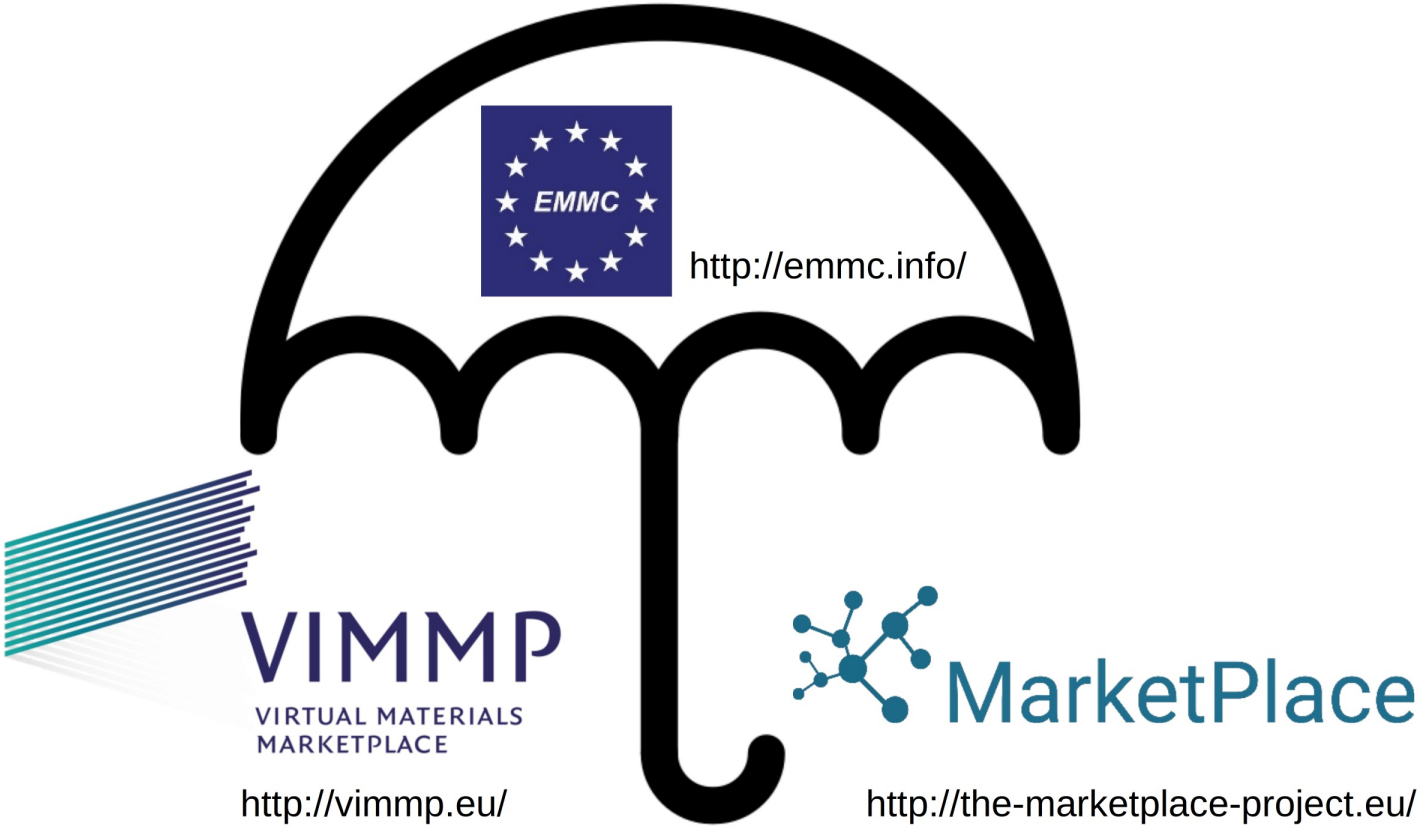
Martin Horsch, Silvia Chiacchiera,
Michael Seaton, and Ilian Todorov
STFC Daresbury Laboratory
UK Research and Innovation

Ontology development at the marketplace level in VIMMP

VIMMP webinar with support from EMMC and MarketPlace

23rd May 2019
Daresbury

VIMMP, MarketPlace, and EMMC-CSA projects



VIMMP ontology webinar

VIMMP
VIRTUAL MATERIALS MARKETPLACE

To subscribe for announcements, send an e-mail to vimmp-webinar@ifam.fraunhofer.de.

EMMC-CSA has recently given a webinar on ontology, provided by Alexandra Simperler.

MarketPlace

Jesper Friis will represent MarketPlace today with a brief statement.

Suggestions for topics to cover in future VIMMP or joint webinars would be welcome.

VIMMP – Virtual Materials Marketplace

- H2020 Project
  - Virtual Materials Marketplace VIMMP (GA 760907)
  - H2020 (NMBP-25-2017)
  - 4 years project – started on 01.01.2018

- **User-friendly hub** to provide:
  - effective use of materials modelling by a wide range of stakeholders
  - facilitating an accelerated speed of development and market deployment of new materials

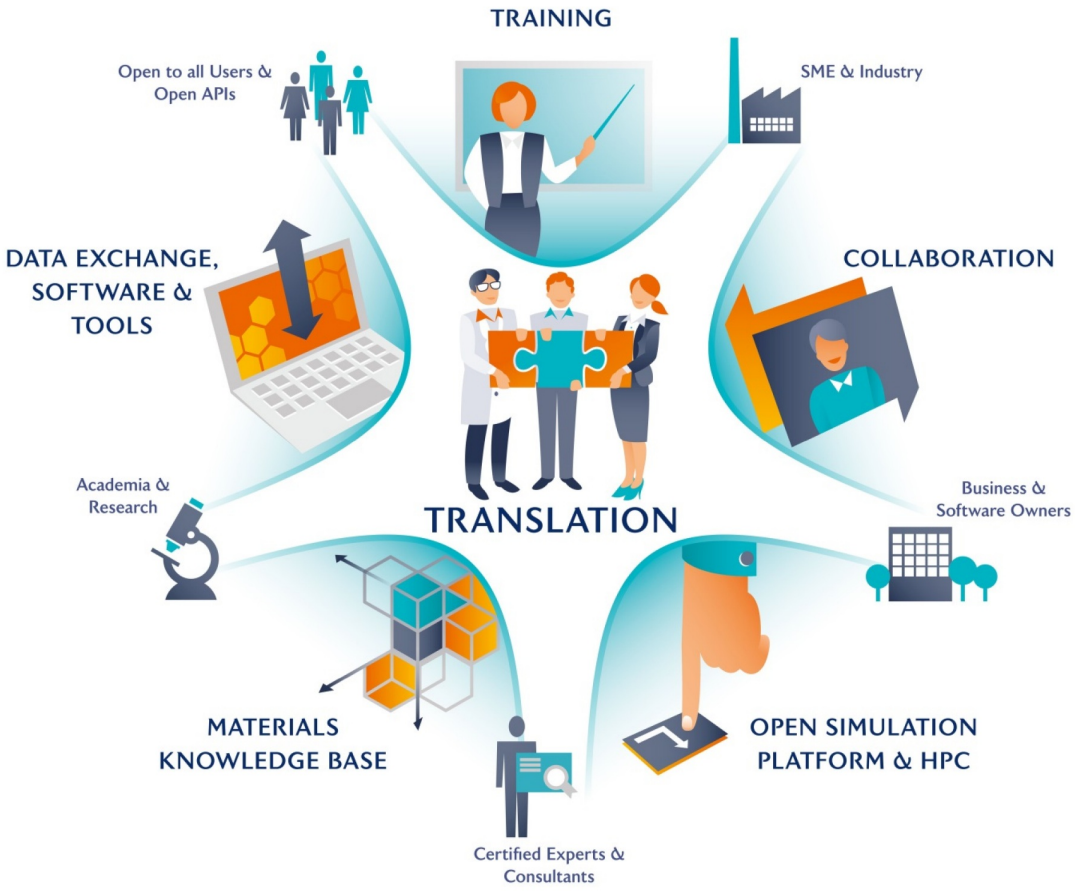
VIMMP – Virtual Materials Marketplace

VIMMP Project Meeting at Fraunhofer IFAM
Bremen, 10.10.2018 to 12.10.2018

VIMMP
VIRTUAL MATERIALS MARKETPLACE

This Project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 760907.

Coordinated by Fraunhofer IFAM



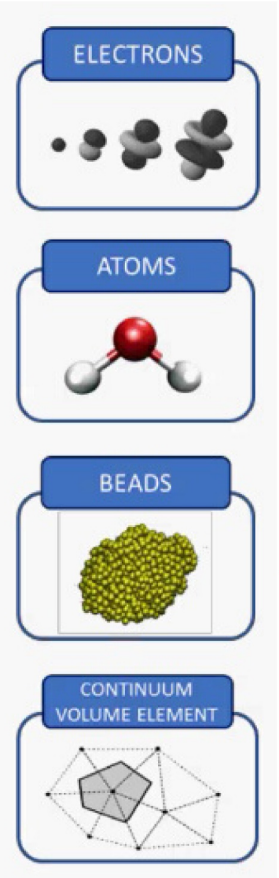
VIMMP will facilitate the **translation** of a **scientific problem** into **modelling work-flows** ready for **simulation** using a range of software tools integrated into an open simulation platform and deployed on cloud services.

Community-driven materials modelling standardization

Time line of EMMC guided semantic-asset development

RoMM VI

Semi-formalized terminology or vocabulary



electronic

atomistic

mesoscopic

continuum

Community-driven materials modelling standardization

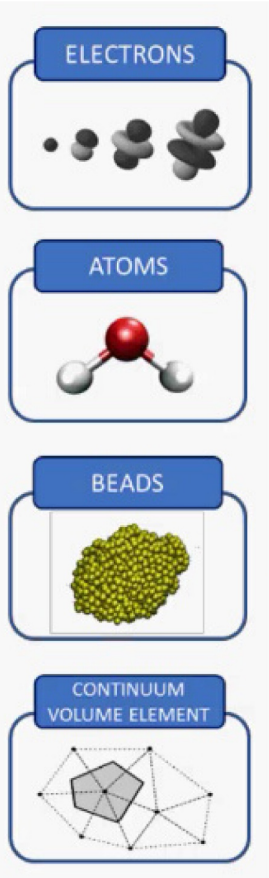
Time line of EMMC guided semantic-asset development

RoMM VI **MODA** **CWA 17284**

Semi-formalized terminology or vocabulary

MODA workflow graph language

CEN European standard



Community-driven materials modelling standardization

Time line of EMMC guided semantic-asset development

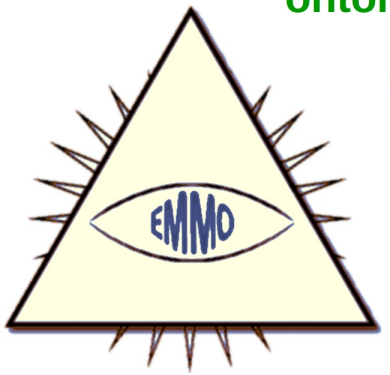
RoMM VI → MODA → CWA 17284 → Ontologies

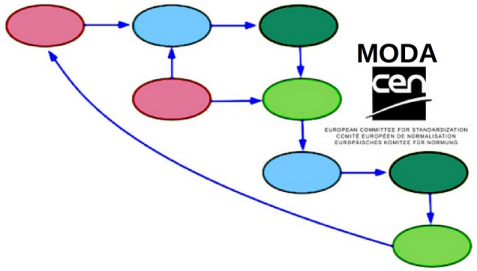
Semi-formalized terminology or vocabulary

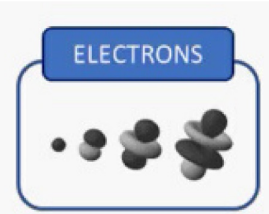
MODA workflow graph language

CEN European standard

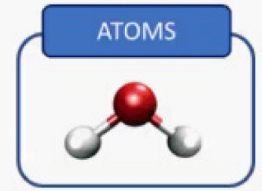
EMMO, EVMPO, marketplace-level, and subdomain-specific ontologies



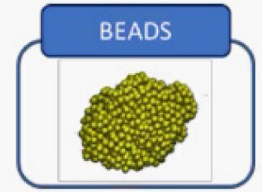




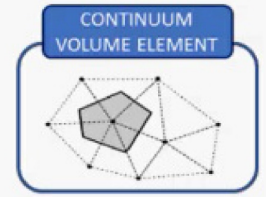
electronic



atomistic



mesoscopic



continuum

European Virtual Marketplace Ontology

The EVMPO provides a structure for the marketplace-level ontologies by formulating **fundamental paradigmatic categories** that correspond to irreducible terms which are seen as constitutive to the virtual-marketplace paradigm.

Recommendation: Any ontology at the marketplace level should follow the structure given by these categories as closely as possible.

VIMMP
VIRTUAL MATERIALS MARKETPLACE
MarketPlace

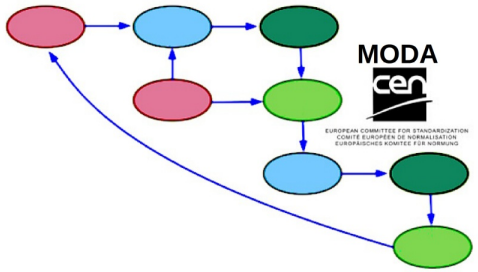
Fundamental paradigmatic categories:

(1) **assessment**, i.e., proposition on accuracy, performance of an entity, or of an entity's trust in another entity
(2) **calendar_event**, i.e., meeting or activity that is scheduled or can be scheduled, equivalent to Vevent from ICALTZD
(3) **communication**, i.e., statement or sequence of statements that can be communicated at a virtual marketplace
(4) **information_content_entity** as defined in the Information Artifact Ontology (IAO)
(5) **infrastructure**, i.e., virtual-marketplace infrastructure (e.g., data access, hardware, and software)
(6) **material** as defined in the European Materials Modelling Ontology (EMMO)
(7) **model**, i.e., entity that can be described by the 2nd section of MODA, equivalent to "model" from the EMMO
(8) **process**, i.e., temporal evolution of one or multiple entities
(9) **product**, i.e., good or service that can be offered either at a virtual marketplace or off-site
(10) **property** as defined in the EMMO
(11) **role** as defined in the EMMO
(12) **simulation**, i.e., a simulation workflow (as in MODA)

Ontologies for the Virtual Materials Marketplace (VIMMP)

MODA Graph Language, CEN Workshop Agreement 17284, and EMMO (Ghedini *et al.*)

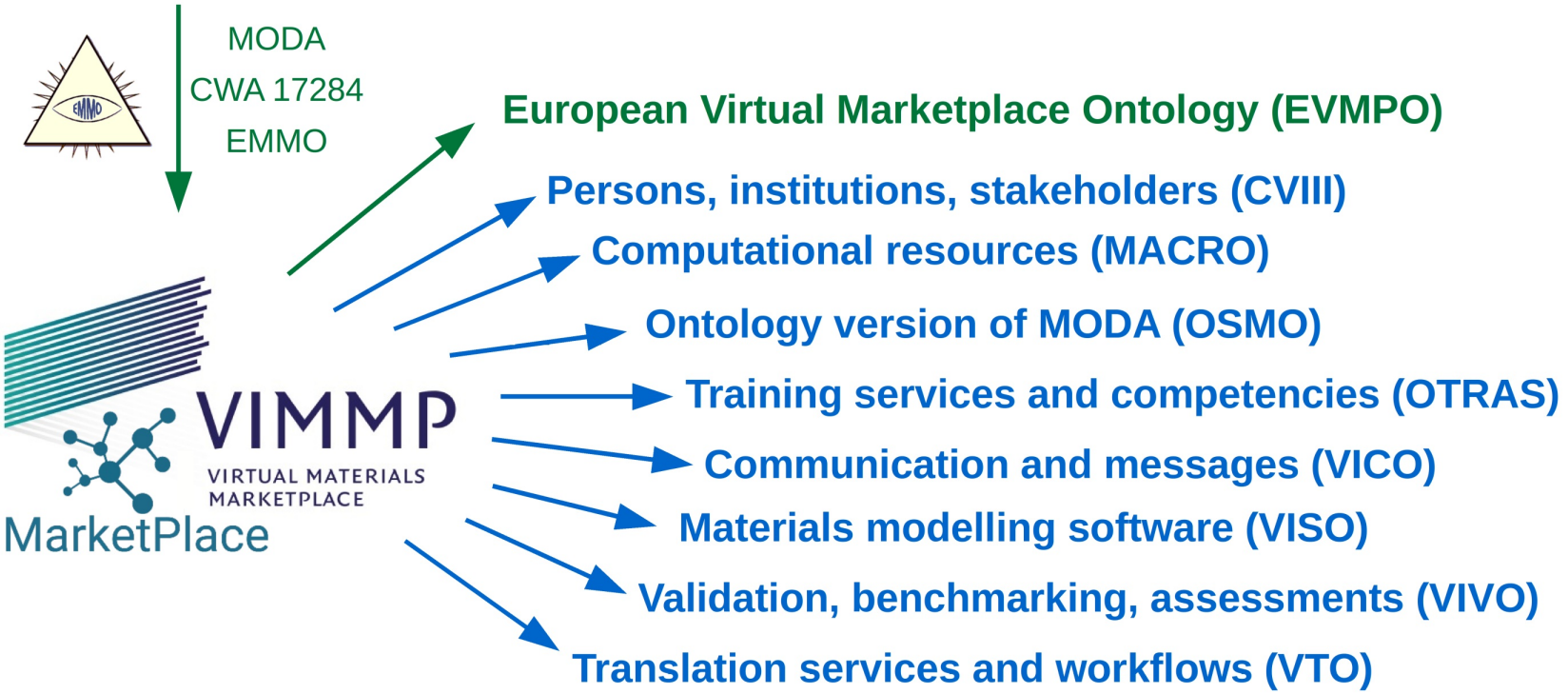
European Virtual Marketplace Ontology (EVMPO)



- **Upper level: EMMO extended by European Virtual Marketplace Ontology (EVMPO)**

Ontologies for the Virtual Materials Marketplace (VIMMP)

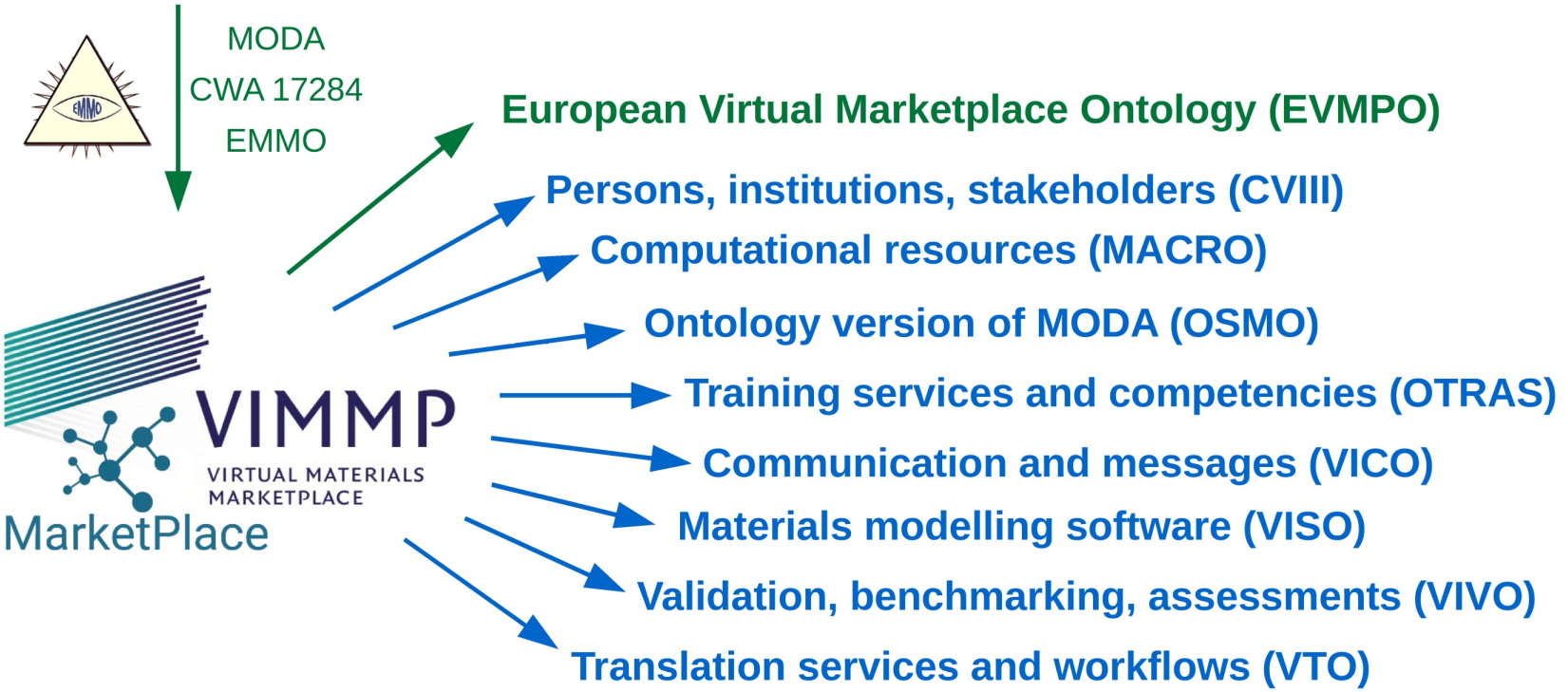
MODA Graph Language, CEN Workshop Agreement 17284, and EMMO (Ghedini *et al.*)



- **Upper level: EMMO extended by European Virtual Marketplace Ontology (EVMPO)**
- **Marketplace-level ontologies: VIMMP in coordination with the MarketPlace project**

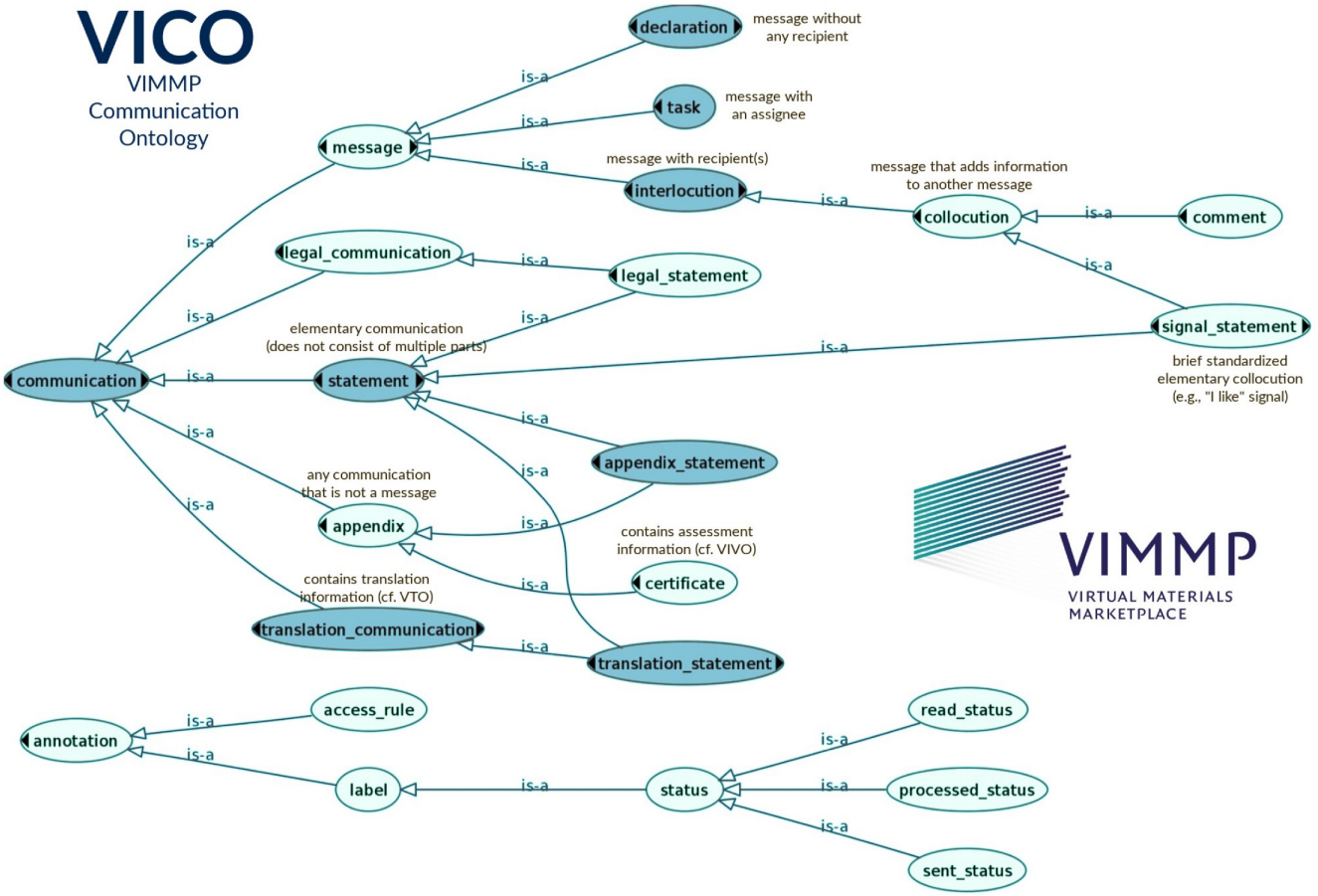
Ontologies for the Virtual Materials Marketplace (VIMMP)

MODA Graph Language, CEN Workshop Agreement 17284, and EMMO (Ghedini *et al.*)



- **Upper level: EMMO extended by European Virtual Marketplace Ontology (EVMPO)**
- **Marketplace-level ontologies: VIMMP in coordination with the MarketPlace project**
- **Subdomain-specific level: Modelling approaches (particle-based, continuum, ...)**

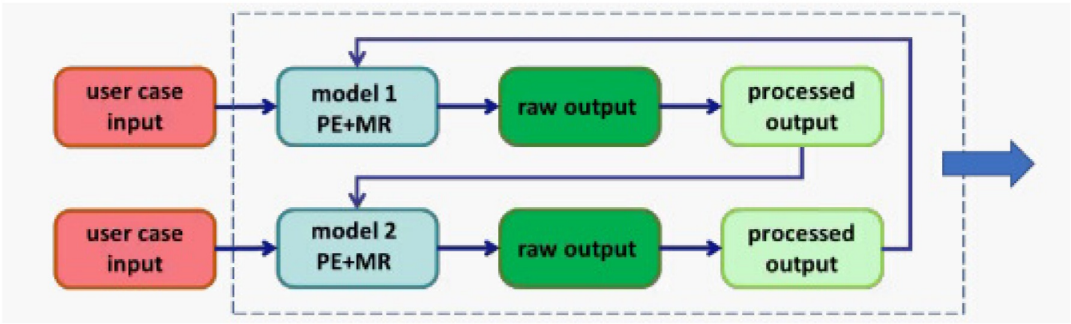
VIMMP Communication Ontology

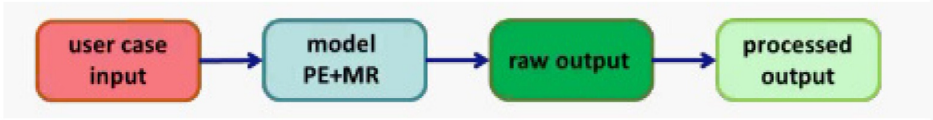


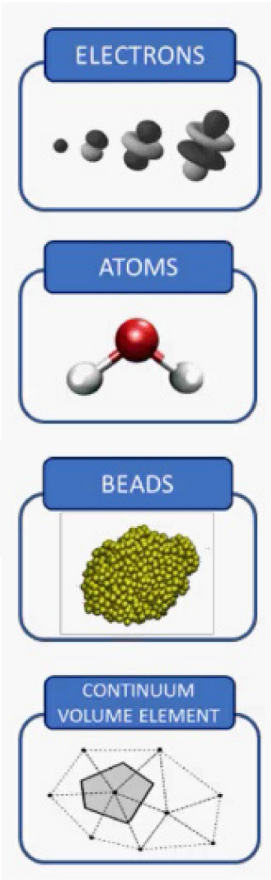
Virtual-marketplace stakeholder communication is formalized by VICO, taking into account specific requirements related to assessment and validation (VIVO) and translation (VTO).

Graph notation for simulation workflow semantics

MODA ("Model data") workflow graph language (CEN standard by CWA 17284)







electronic

atomistic

mesoscopic

continuum

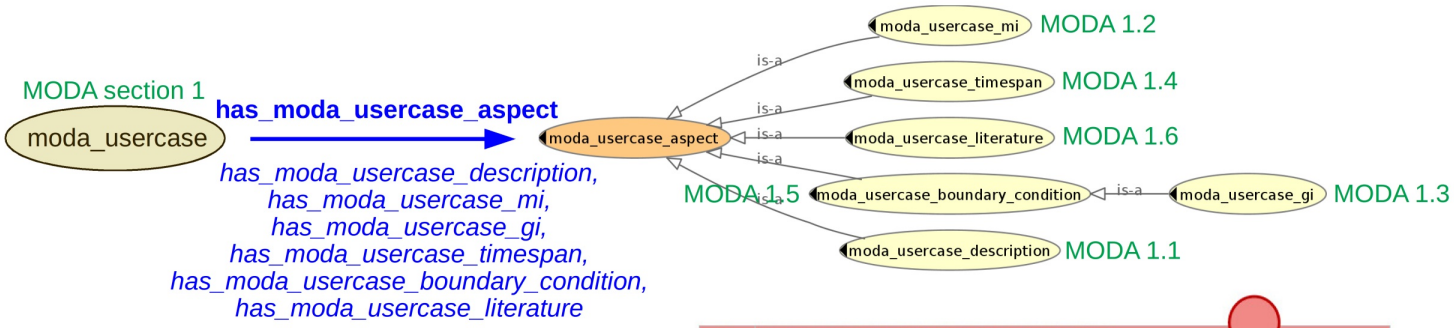
MODA

(1) User Case
(2) Model
(3) Solver
(4) Processor

cen

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

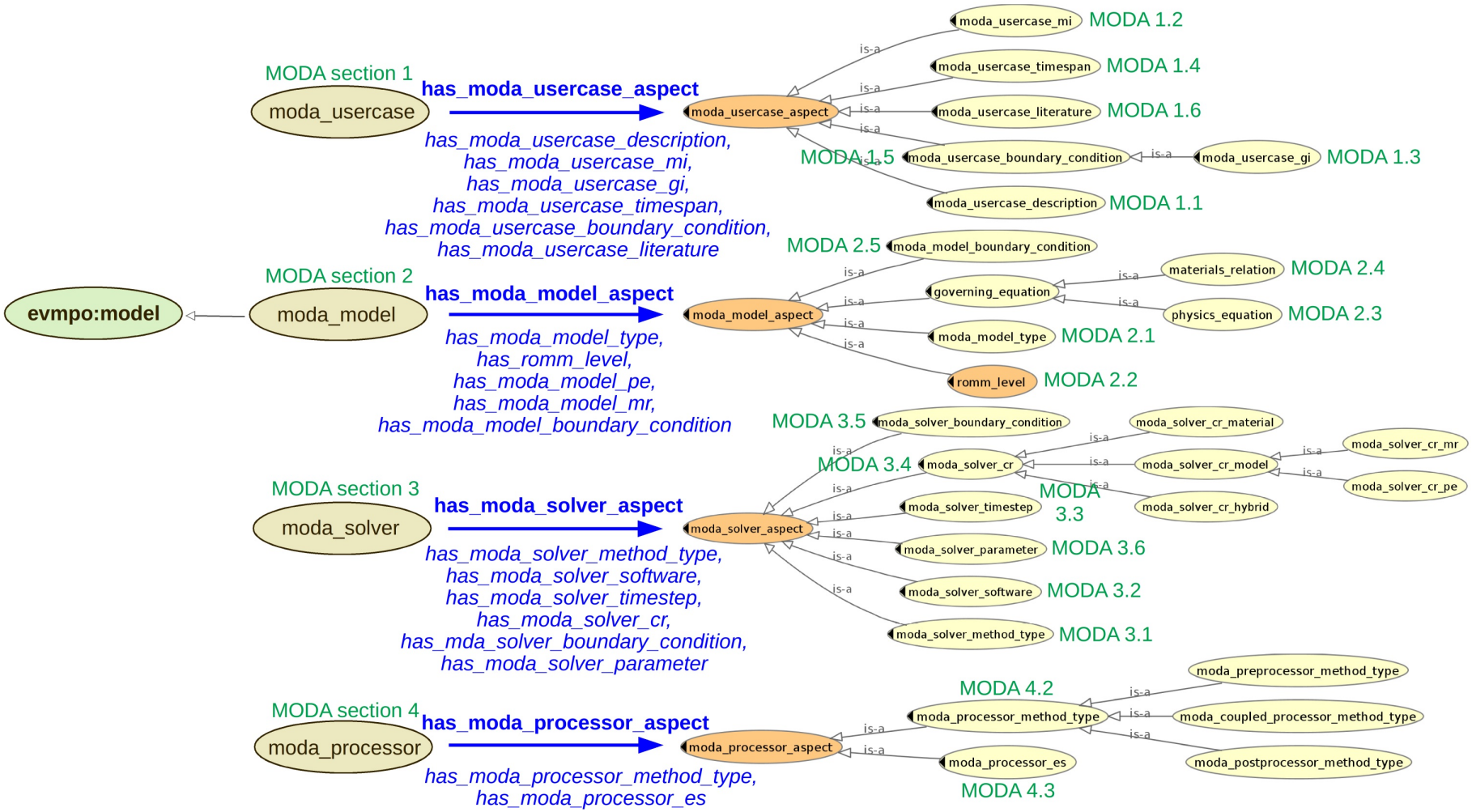
MODA section structure: Ontological representation



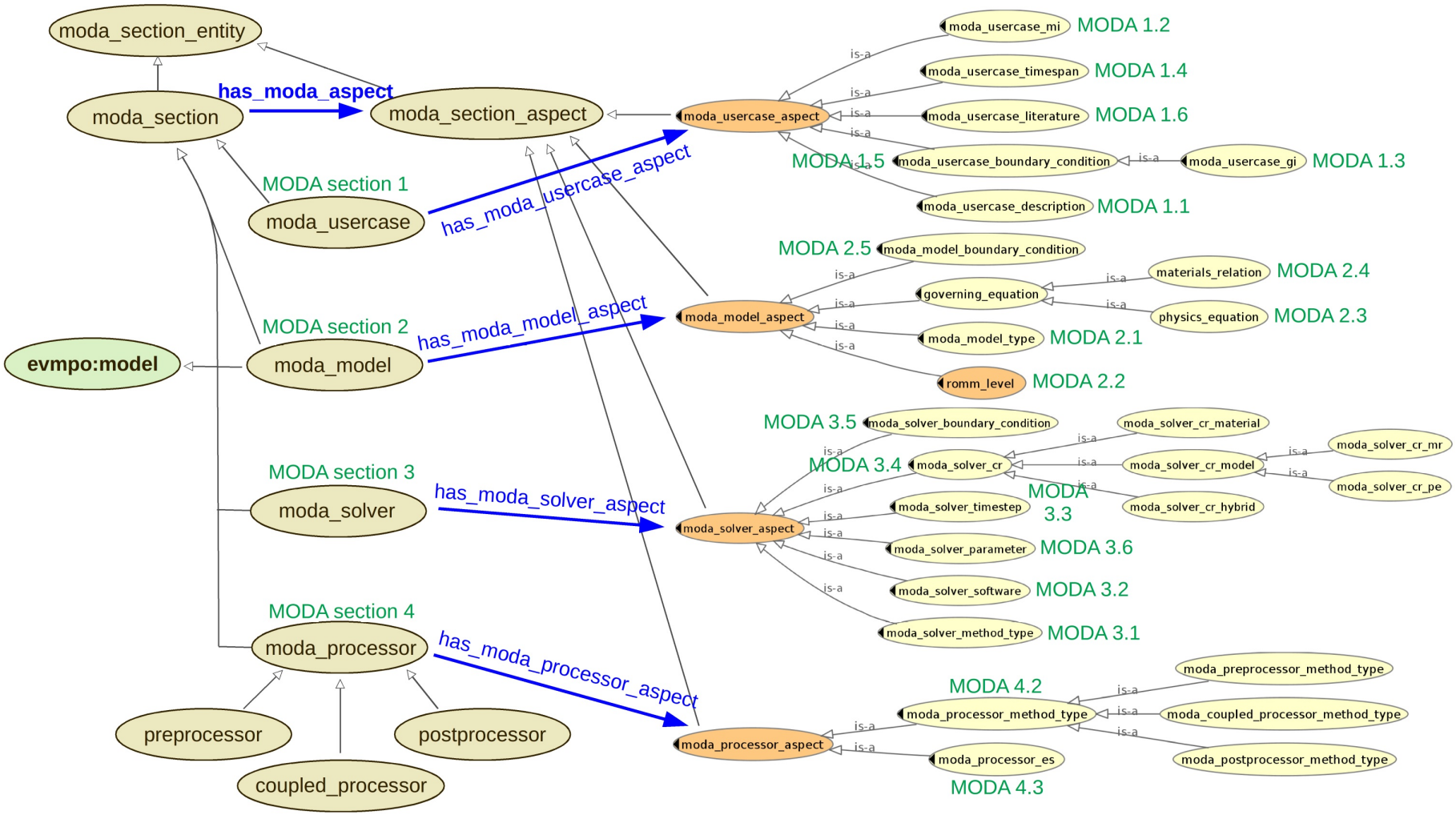
1	Aspect of the User Case/System to be Simulated	
1.1	Aspect of the User Case to be Simulated	*Describe the aspects of the user case textually.* *No modelling information should appear in this box. This case could also be simulated by other models in a benchmarking operation!* *The information in this chapter can be end-user information, measured data, library data etc. It will appear in the pink circle of your workflow picture.* *Simulated input which would have been calculated by another model should not be included (but in chapter 2.4)* *Also the result of pre-processing necessary to translate the user case specifications to values for the physics variables of the entities van be documented here.*
1.2	Material	*Describe the chemical composition, ...and the values used for properties and from which database these are taken. If pre-processing was needed please specify the methodology.*
1.3	Geometry	*Size, form, picture of the system (if applicable)* *Note that computational choices like simulation boxes are to be documented in chapter 3.*
1.4	Time Lapse	*Duration of the case to be simulated.* *This is the duration of the situation to be simulated. This is not the same as the computational times to be given in chapter 3.*
1.5	Manufacturing process or in-service conditions	*If relevant, please list the conditions to be simulated (if applicable).* *These can be boundary, initial and global conditions.* *E.g. heated walls, external pressures and bending forces.* *Please note that these might appear as terms in the PE or as boundary conditions, and this will be documented in the relevant chapters.* *Note: These conditions will be expressed in physics relations in Ch 2.4* *Please specify the values used for parameters and from which database these are taken. If pre-processing was needed please specify the methodology.*
1.6	Publication on this data	*Publication documenting the simulation with this single model (if available and if not already included in the overall publication).*

(1) User Case
(2) Model
(3) Solver
(4) Processor

MODA section structure: Ontological representation



MODA section structure: Ontological representation



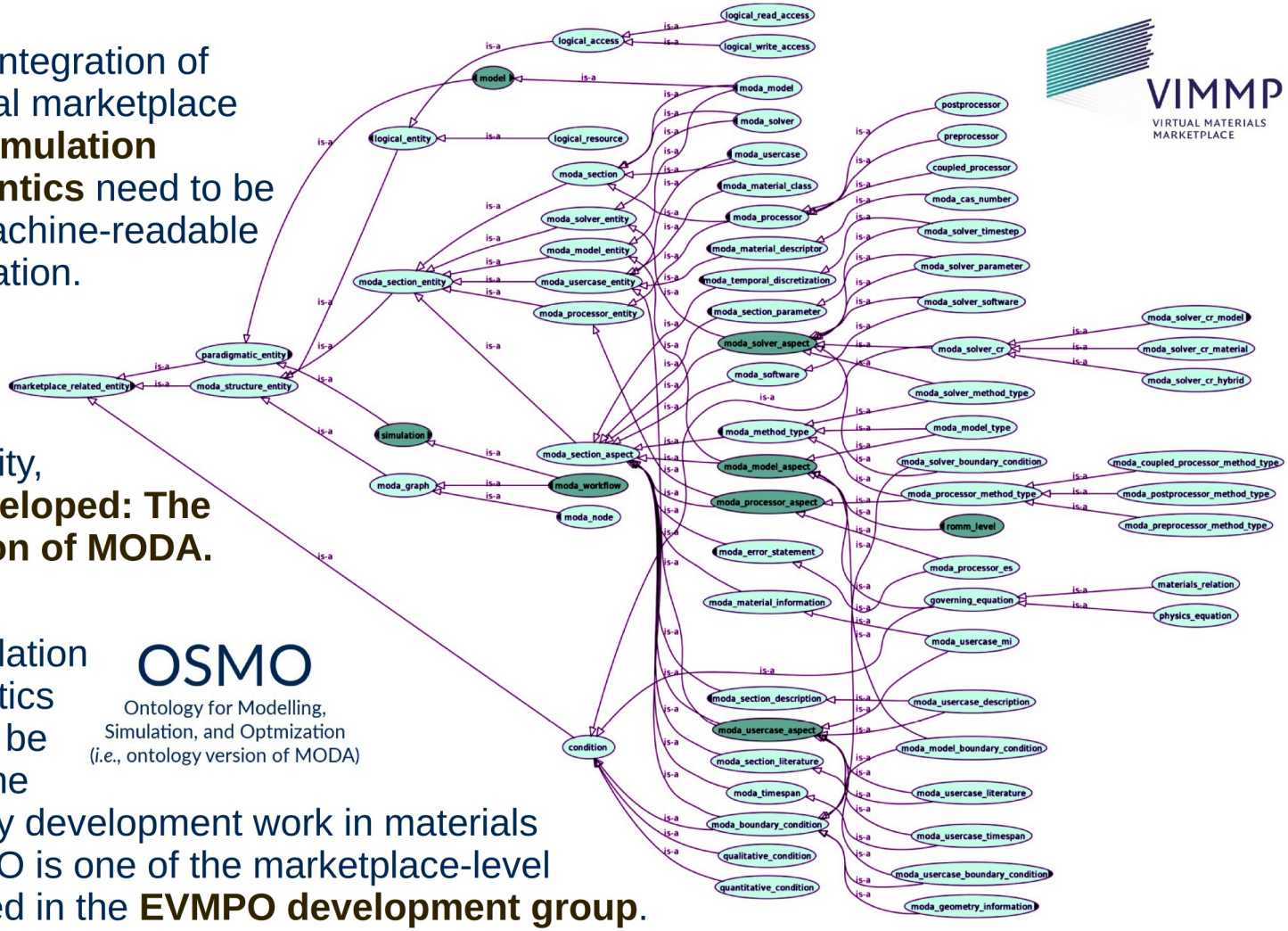
Ontology for Simulation, Modelling, and Optimization

To facilitate the integration of MODA into virtual marketplace infrastructure, **simulation workflow semantics** need to be provided at a machine-readable level of formalization.

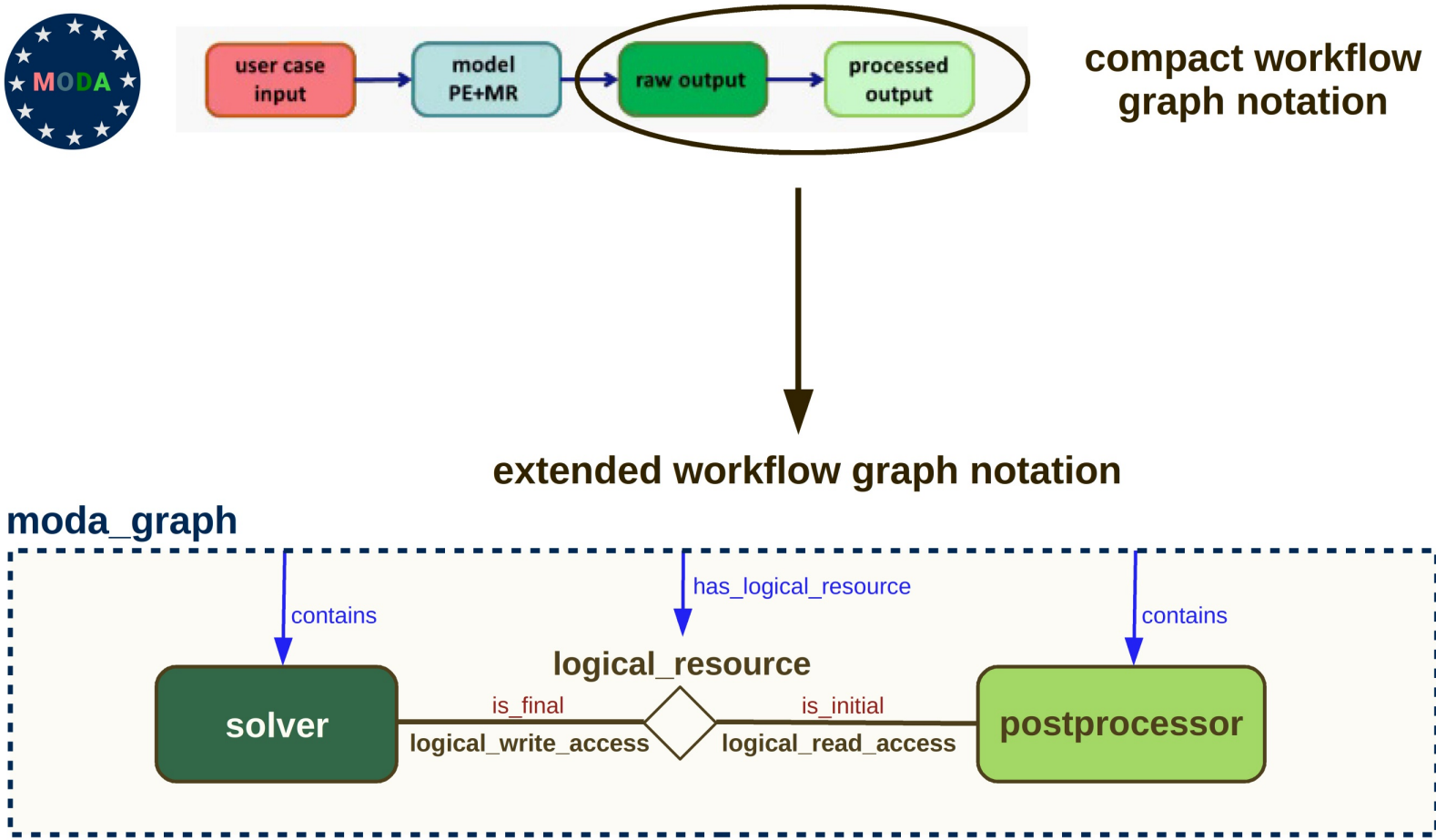
Following the approach of the EMMC community, **OSMO was developed: The ontology version of MODA.**

By OSMO, simulation workflow semantics from MODA can be integrated into the ongoing ontology development work in materials modelling. OSMO is one of the marketplace-level ontologies shared in the **EVMPO development group**.

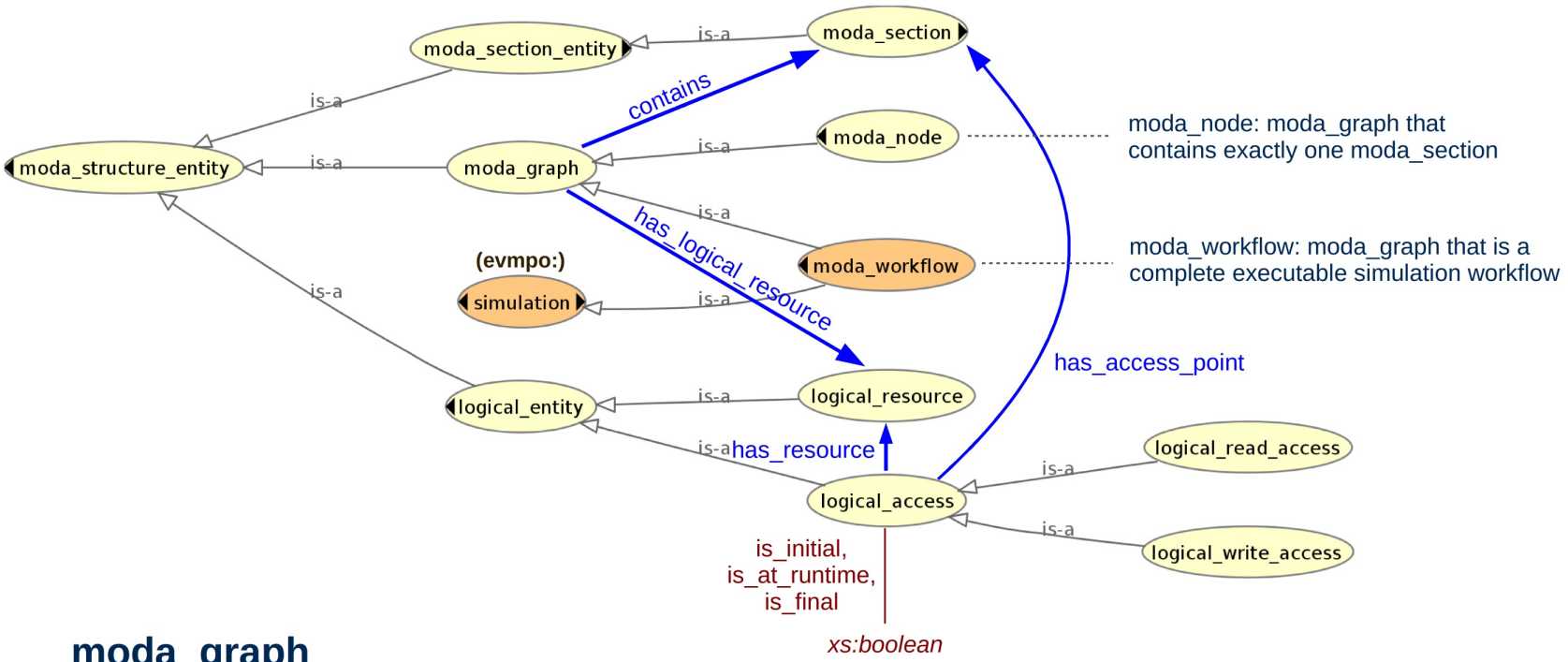
OSMO
Ontology for Modelling, Simulation, and Optmization (*i.e.*, ontology version of MODA)

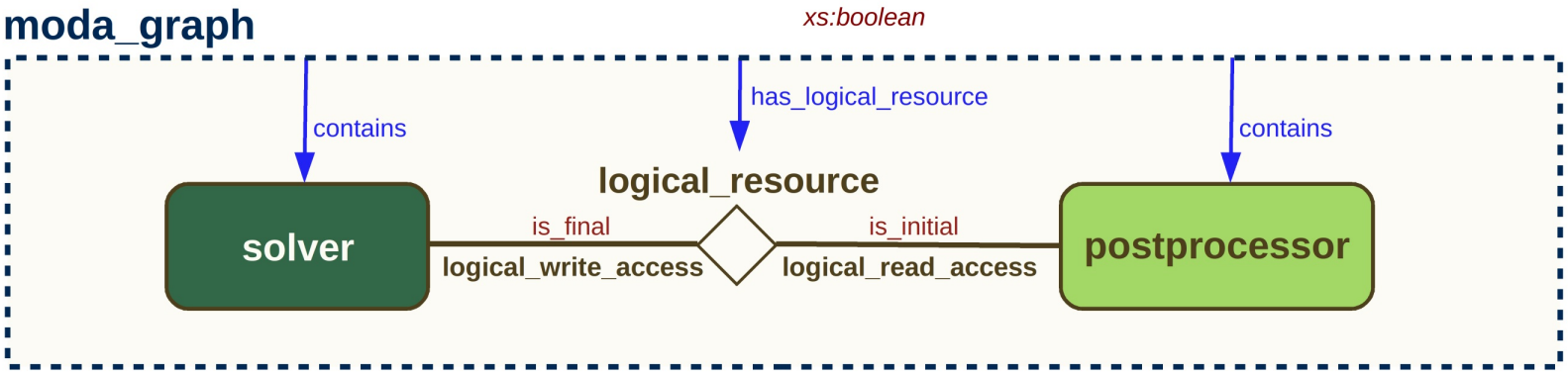


Simulation workflows and related entities in OSMO

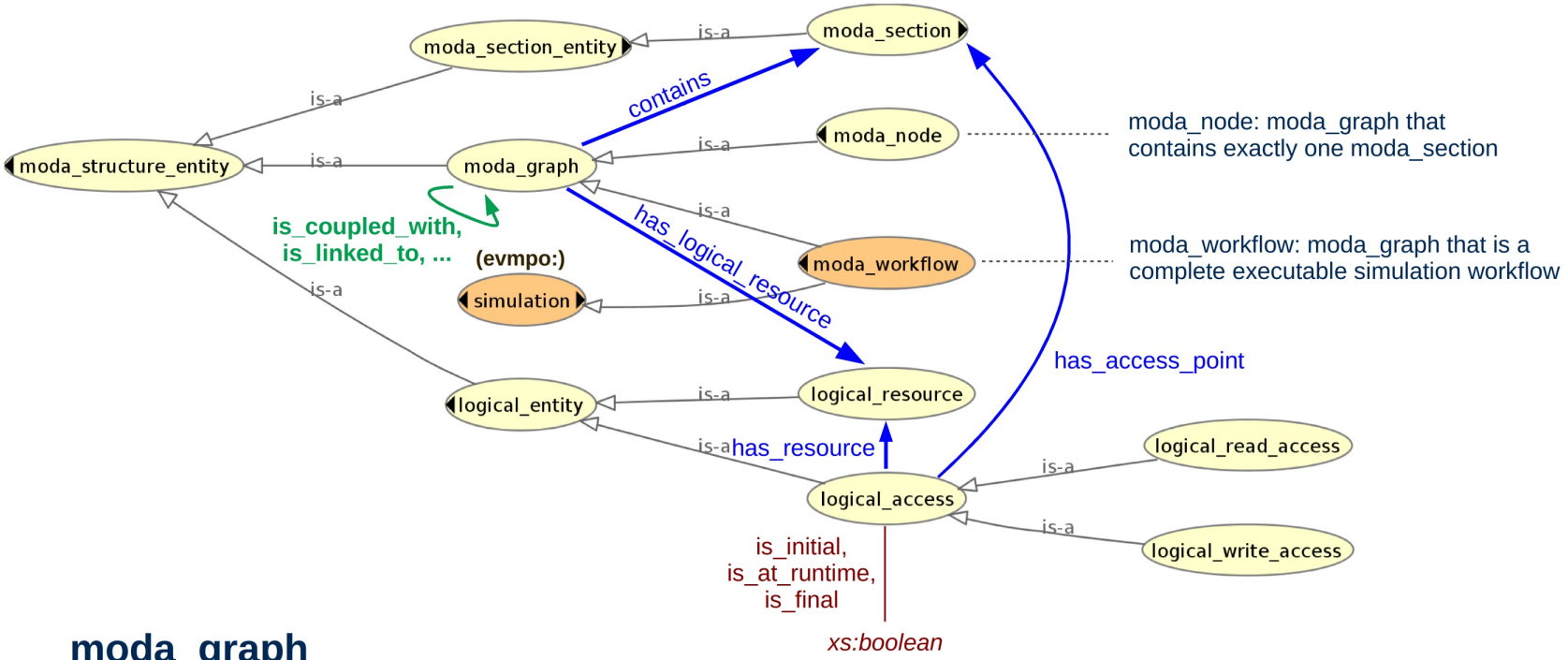


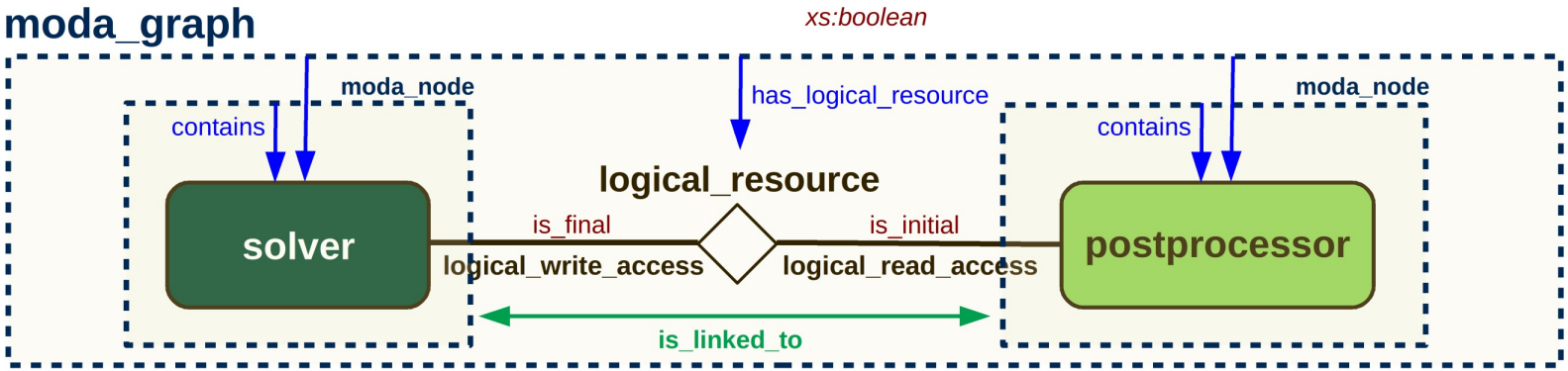
Simulation workflows and related entities in OSMO





Simulation workflows and related entities in OSMO



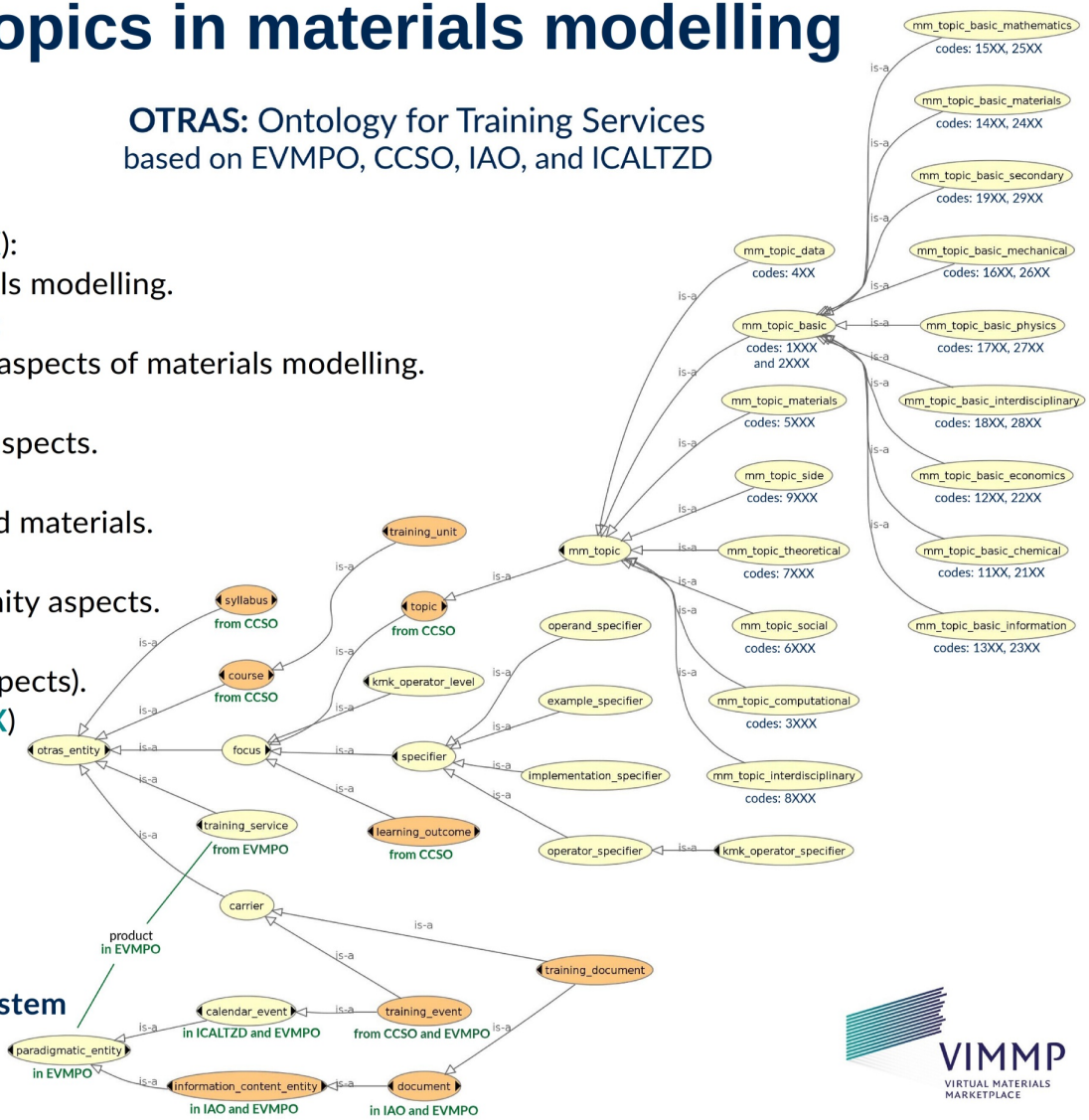


Training ontology: Topics in materials modelling

The training ontology will include **topic** and **operator** catalogues.

OTRAS: Ontology for Training Services
based on EVMPO, CCSO, IAO, and ICALTZD

mm_topic_basic (codes **1XXX** and **2XXX**):
Basic prerequisites for materials modelling.
mm_topic_computational (codes **3XXX**):
Computational and numerical aspects of materials modelling.
mm_topic_data (codes **4XXX**):
Data science and technology aspects.
mm_topic_materials (codes **5XXX**):
Topics related to fluid and solid materials.
mm_topic_social (codes **6XXX**):
Social, economic, and community aspects.
mm_topic_theoretical (codes **7XXX**):
Theory (non-computational aspects).
mm_topic_interdisciplinary (codes **8XXX**)
mm_topic_side (codes **9XXX**):
Topics from other disciplines



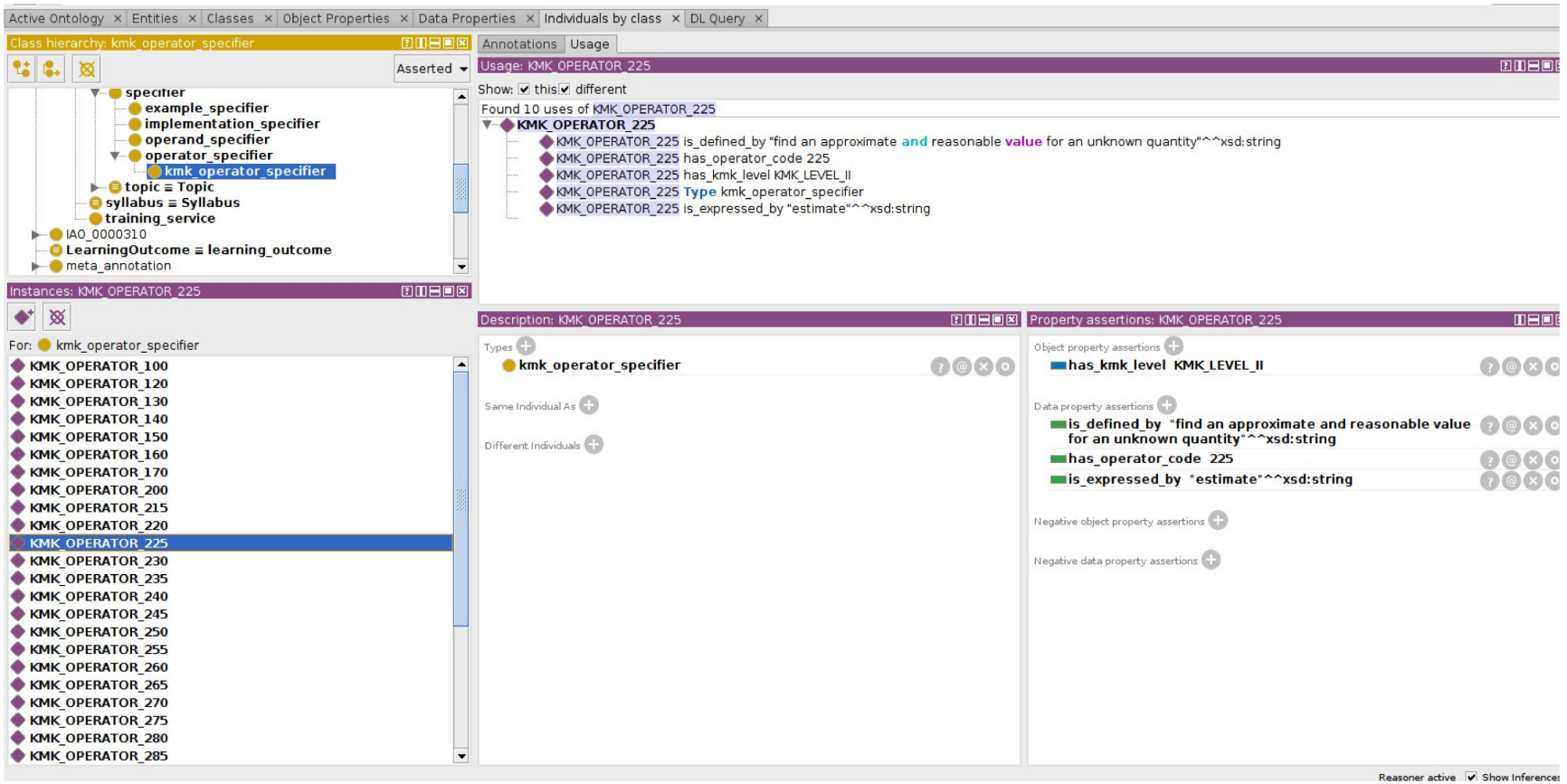
Below, possible use of:

- **AIP Physics and Astronomy Classification Scheme (PACS)**
- **ACM Computing Classification System**
- **Springer Verlag semantic assets**
- **Dewey Decimal Classification**

VIMMP
VIRTUAL MATERIALS MARKETPLACE

Training ontology: Operators for learning outcomes

"After successfully completing X_1, **participants can X_2** with respect to X_3 by doing X_4; for example, X_5." (Note: X_4 and X_5 are not required, and X_1 is not an outcome.)



Training ontology: Operators (KMK catalogue)

"After successfully completing X_1, **participants can X_2** with respect to X_3 by doing X_4; for example, X_5." (Note: X_4 and X_5 are not required, and X_1 is not an outcome.)

1XX – Operators for **basic competencies:** "to name/label" (code **120**), "to outline/present" (code **130**), "to list/give" (code **140**), "to write a lab report/data log" (code **150**), "to sketch" (code **160**), "to draw" (code **170**).

2XX – Operators for **intermediate competencies:** "to compare" (code **215**), "to deduce" (code **220**), "to estimate" (code **225**), "to analyse and identify" (code **230**), "to apply" (code **235**), "to calculate" (code **240**), "to describe" (code **245**), "to find" (code **250**), "to explain" (code **255**), "to describe and explain" (code **260**), "to formulate" (code **265**), "to derive" (code **270**), "to sort/group/classify" (code **275**), "to test/verify" (code **280**), "to investigate/examine" (code **285**), "to generalize" (code **290**), "to summarize" (code **295**).

3XX – Operators for **advanced competencies:** "to propose a hypothesis" (code **320**), "to evaluate" (code **330**), "to justify/give reasons" (code **340**), "to comment on/assess" (code **350**), "to prove" (code **360**), "to discuss" (code **370**), "to interpret" (code **380**), "to plan" (code **390**).

Training ontology: Operators (extended KMK catalogue)

"After successfully completing **X_1**, **participants can X_2** with respect to **X_3** by doing **X_4**; for example, **X_5**." (Note: **X_4** and **X_5** are not required, and **X_1** is not an outcome.)

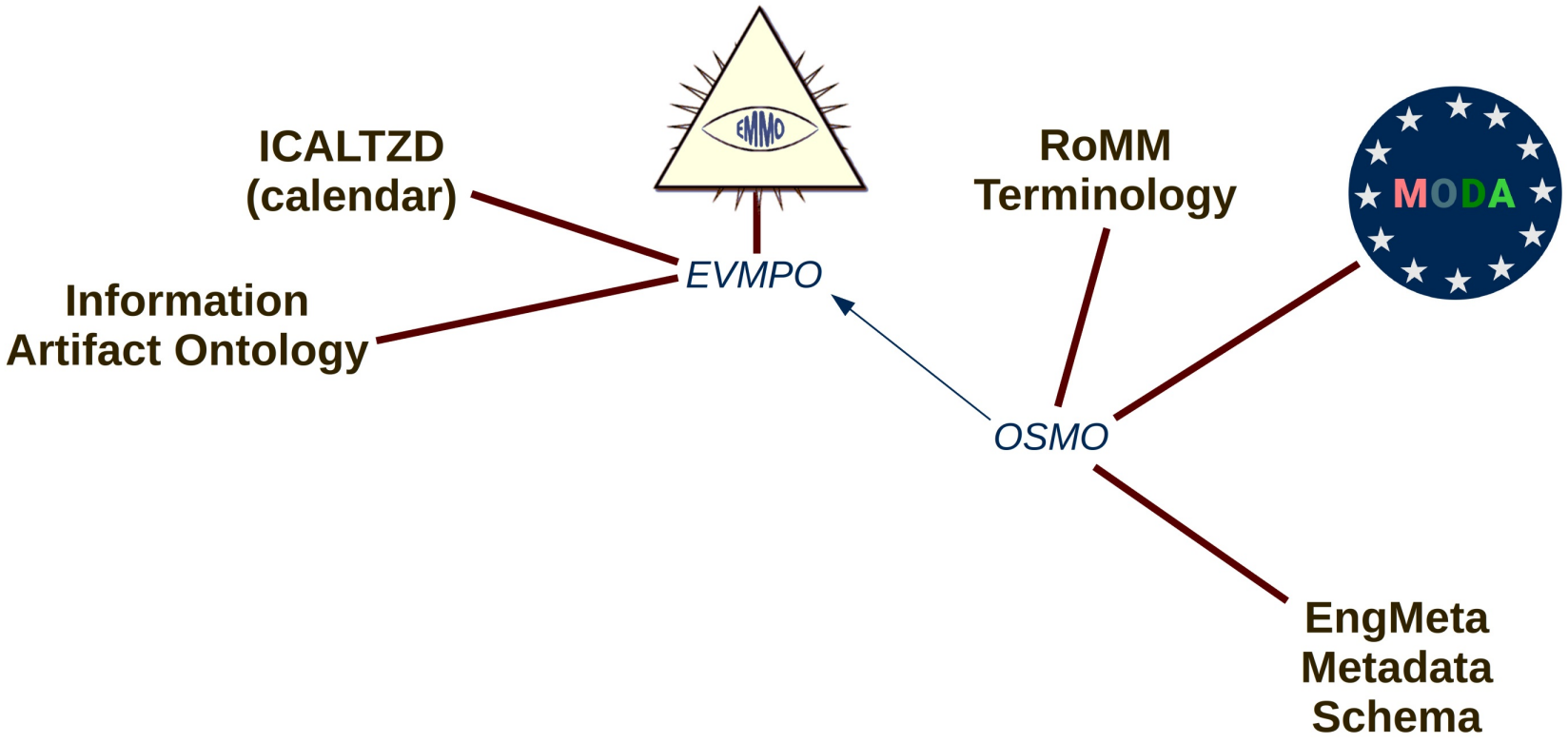
1XX – Operators for **basic competencies:** "to name/label" (code **120**), "to outline/present" (code **130**), "to list/give" (code **140**), "to write a lab report/data log" (code **150**), "to sketch" (code **160**), "to draw" (code **170**).

2XX – Operators for **intermediate competencies:** "to compare" (code **215**), "to deduce" (code **220**), "to estimate" (code **225**), "to analyse and identify" (code **230**), "to apply" (code **235**), "to calculate" (code **240**), …, "to summarize" (code **295**).

3XX – Operators for **advanced competencies:** "to propose a hypothesis" (code **320**), "to evaluate" (code **330**), "to justify/give reasons" (code **340**), "to comment on/assess" (code **350**), "to prove" (code **360**), "to discuss" (code **370**), …, "to plan" (code **390**).

4XX – Operators for **expert competencies:** "to review/evaluate critically" (code **420**), "to advise/manage" (code **425**), "to characterize experimentally" (code **430**), "to document" (code **435**), "to carry out professionally" (code **440**), "to correspond" (code **445**), "to teach" (code **450**), "to plan/project" (code **455**), "to conduct an exam" (code **460**), "to systematize" (code **465**), "to expand/extend/generalize" (code **470**), "to simplify/reduce" (code **475**), "to innovate/develop" (code **480**).

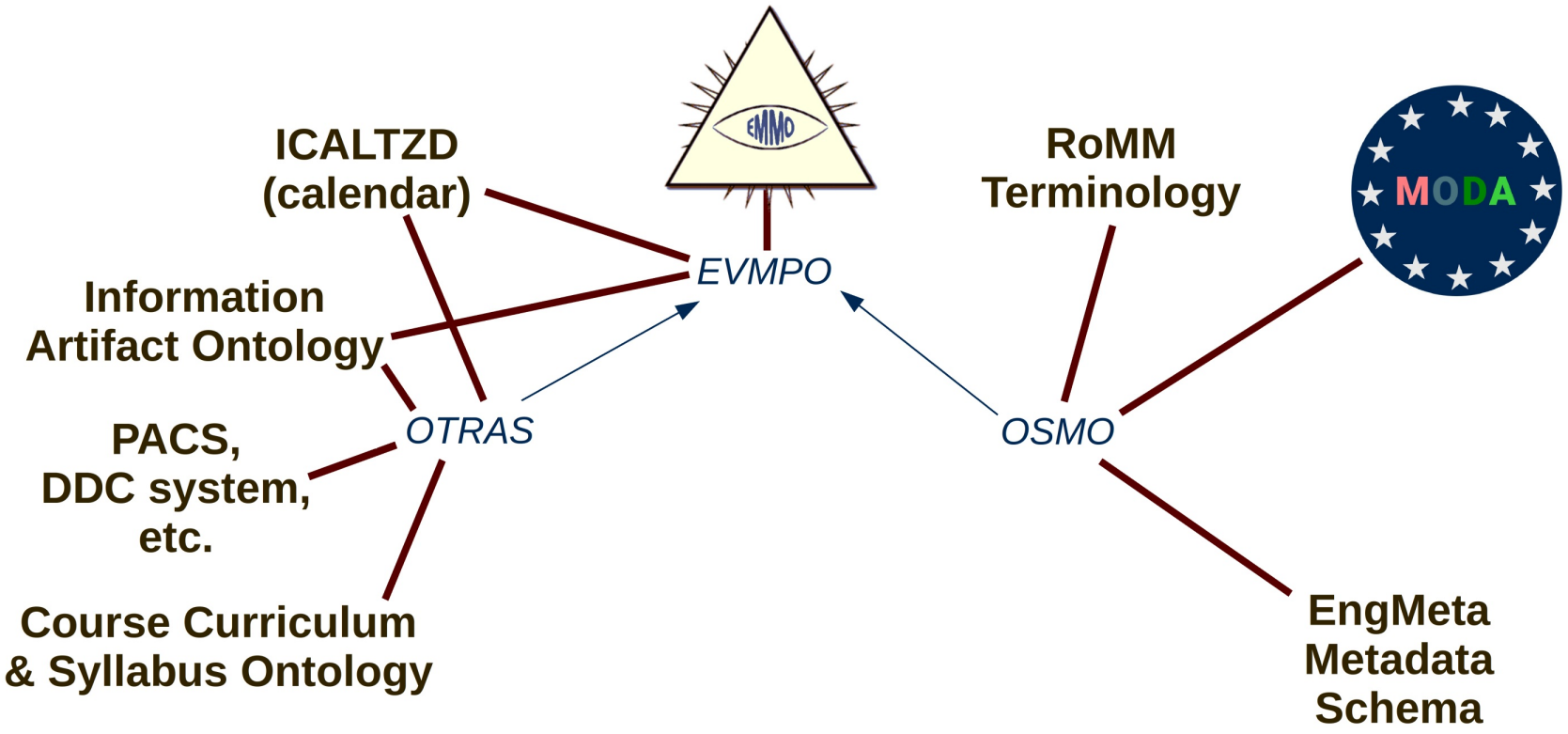
Semantic asset architecture at marketplace & upper level



Blue: Own semantic assets

Black: Related external semantic assets with an overlap or interaction

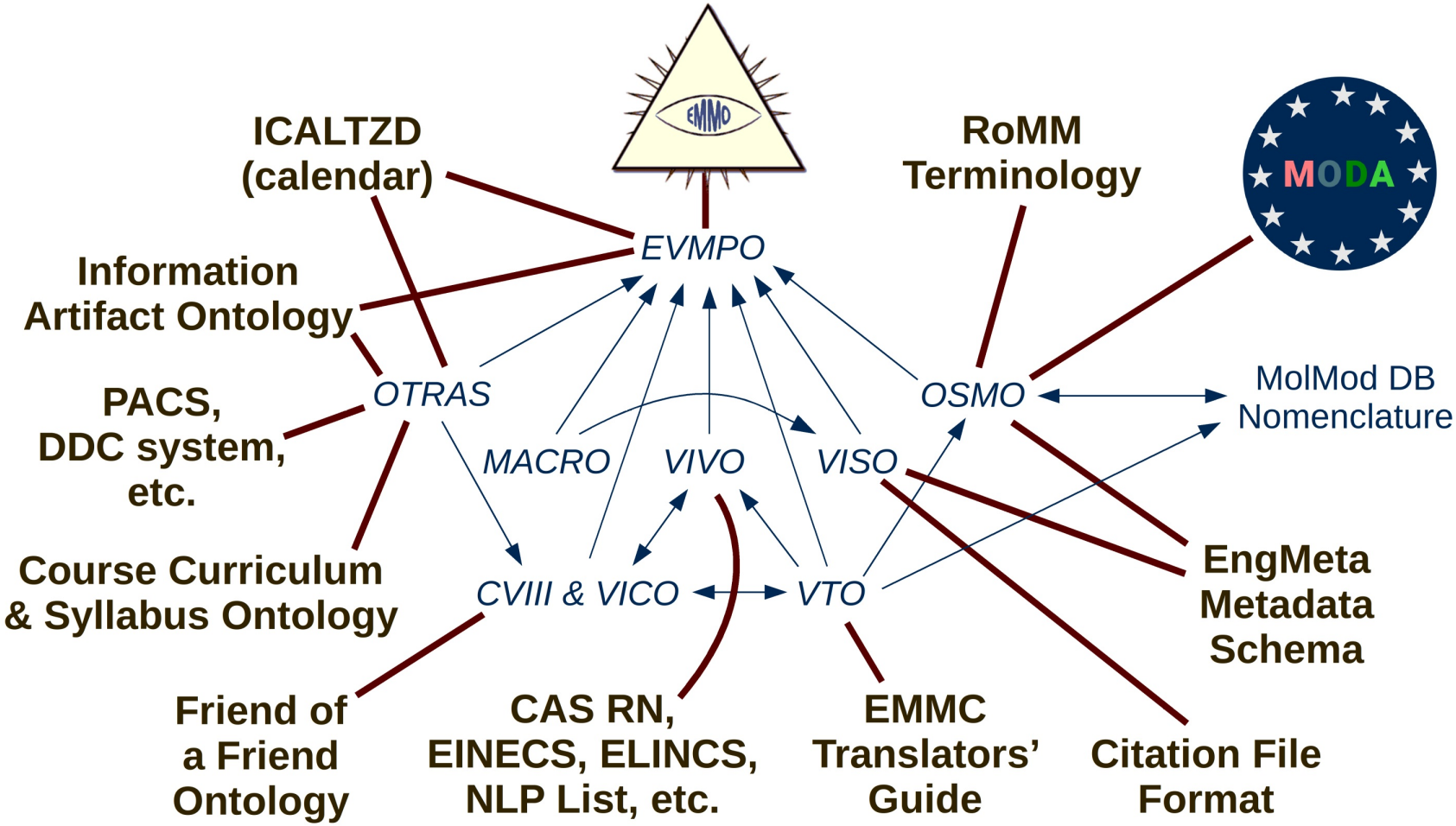
Semantic asset architecture at marketplace & upper level



Blue: Own semantic assets

Black: Related external semantic assets with an overlap or interaction

Semantic asset architecture at marketplace & upper level



Blue: Own semantic assets

Black: Related external semantic assets with an overlap or interaction

Representation language and implementation

TTL format applied to OWL DL

Terse Triple Language (TTL) consists of “triples”:

```
subject    a                  class_of_subject;
           has_property       object;
           also_has_property  other_object.
```

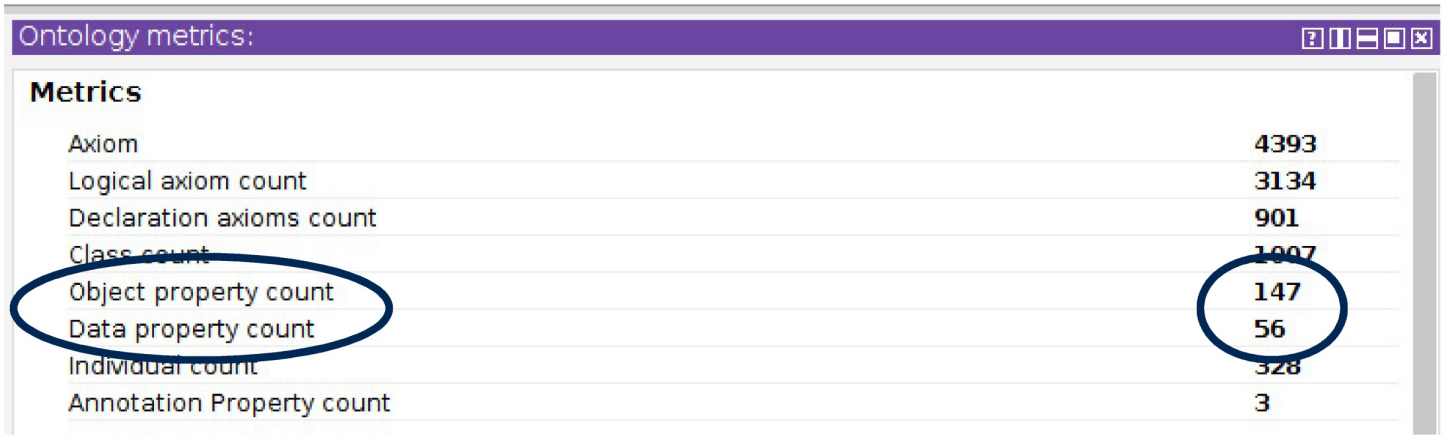
Example

```
osmo:logical_access a owl:Class;
  rdfs:subClassOf osmo:logical_entity;
  rdfs:comment "connection by which a logical resource is accessed".
osmo:op_logical_access a owl:Class;
  rdfs:subClassOf owl:ObjectProperty.

osmo:has_access_point a osmo:op_logical_access;
  rdfs:domain osmo:logical_access;
  rdfs:range osmo:moda_section.
osmo:has_resource a osmo:op_logical_access;
  rdfs:domain osmo:logical_access;
  rdfs:range osmo:logical_resource;
  rdfs:subPropertyOf evmpo:has_meta_annotation.

osmo:logical_access
  owl:disjointUnionOf
    (osmo:logical_read_access osmo:logical_write_access).
```

Relations and objects in ontologies



The European Virtual Marketplace Ontology, and the marketplace-level ontologies connected to it, define over 200 relations between objects and objects (owl:ObjectProperty) and between objects and elementary data (owl:DatatypeProperty).

The paradigm inherited from the European Materials Modelling Ontology distinguishes three main types of object properties:

- **membership** (set theory)
  - **representation** (semiotics)
    - **parthood and slicing** (mereotopology)

Relations and objects in ontologies ./. object orientation

Inheritance of relations (methods) for multiple subclasses with a common superclass

method p_A explitiely defined for class *A*
method p_B explitiely defined for class *B*
method p_C explitiely defined for class *C*

… in object oriented programming

all objects of class *A* have the method p_A
all objects of class *B* are objects of class *A*
all objects of class *C* are objects of class *A*

---

all objects of class *B* have the method p_A
all objects of class *C* have the method p_A

In object oriented programming, subclasses inherit the properties of the superclass.

Relations and objects in ontologies ./. object orientation

Inheritance of relations (methods) for multiple subclasses with a common superclass

relation p_A explitiсly defined for class *A*
relation p_B explitiсly defined for class *B*
relation p_C explitiсly defined for class *C*

… in object oriented programming

all objects of class *A* have the method p_A
all objects of class *B* are objects of class *A*
all objects of class *C* are objects of class *A*

---

all objects of class *B* have the method p_A
all objects of class *C* have the method p_A

… in ontologies (“semantic web” concept)

all entities with relation p_B are objects of class *B*
all entities with relation p_C are objects of class *C*
all objects of class *B* are objects of class *A*
all objects of class *C* are objects of class *A*

---

all entities with relation p_B are objects of class *A*
all entities with relation p_C are objects of class *A*

In RDFS and OWL, the superclass inherits the properties of its subclasses.

Relations and objects in ontologies used for data ingest

Inheritance of relations (methods) for multiple subclasses with a common superclass

relation p_A explitiсly defined for class *A*
relation p_B explitiсly defined for class *B*
relation p_C explitiсly defined for class *C*

Approach to achieve congruency between the OOP and semantic web paradigms:

relation p_A is an instance of class op$_A$
relation p_B is an instance of class op$_B$
relation p_C is an instance of class op$_C$

Application during data ingest (a user is asked for data needed to specify an object):

- Scenario: During data ingest, a user wants to specify properties of an instance of class *B*.

Relations and objects in ontologies used for data ingest

Inheritance of relations (methods) for multiple subclasses with a common superclass

relation p_A explitiсly defined for class *A*
relation p_B explitiсly defined for class *B*
relation p_C explitiсly defined for class *C*

Approach to achieve congruency between the OOP and semantic web paradigms:

relation p_A is an instance of class op_A
relation p_B is an instance of class op_B
relation p_C is an instance of class op_C

Application during data ingest (a user is asked for data needed to specify an object):

- Scenario: During data ingest, a user wants to specify properties of an instance of class *B*.
- A triple store containing the ontology can be queried for instances of the class op_B.
- Since op_A is defined as a subclass of op_B, this query yields both p_A and p_B.
- The user is asked the right questions, covering properties defined for class *A* and class *B*.

VISO – VImmp Software Ontology

VISO's main purpose is to describe the software, addressing mostly its capabilities (both model and solver aspects), but also licensing, requirements (as libraries and operating systems) and compatibility[1] with other tools.

It would be used to structure the ingestion of information about software tools on the virtual marketplace. The same keywords will be then available to the users to browse the tools and compare them.

[1] Following E. Ghedini (EMMC), we distinguish between compatibility and interoperability, namely:

compatibility (=ability to exchange information directly, no need to interface)
interoperability (=ability to exchange information through a common language)

Categories at the upper level:

(1) **agent** = An entity (individual, group, institution) that can potentially act on a virtual marketplace [=**cviii:agent**]

(2) **software** = A computer program. Can be a software tool, a compiler, or an operating system.

(3) **license** = Regulation of the right to use, modify and distribute something, in this case software. [Subclass of **evmpo:legal_document**]

(4) **programming_language** = A language that can be used to write software.

(5) **solver_feature** = Capability of a software tool, intended as a numerical algorithm which is implemented.

(6) **model_feature** = Capability of a software tool, intended as a model aspect that can be addressed.

(7) **modelling_related_entity** = High level concept related to modelling, as statistical mechanics, the RoMM models, fundamental physics equation, etc.

(8) **property** = A feature that can be measured or computed [=**emmo:property**]

VISO – VImmp Software Ontology (2)

The main relations between objects in VISO are:

has_feature = To describe the features of a software tool [Inverse: **is_feature_of**]

is_tool_for_model = Relates software tools and RoMM models

is_compatible_with = Asserts compatibility between software tools

is_distributed_by = Relates tools and agents [Inverse: **is_distributor_of**]

has_license = Relates software and license

requires = Relates a software tool to libraries and/or operating systems

can_run_on = Relates a software tool to operating systems

Some relations between objects and literals are:

is_free, **is_open_source**, **is_a_library**, **has_a_gui**

Example for a software tool (extract from a .TTL file):

```
:DL_POLY a viso:software_tool;
  viso:is_free false;
  viso:is_free_to_academic true;
  viso:has_a_gui true;
  viso:is_open_source true;
  viso:is_a_library false;
  viso:is_distributed_by :STFC;
  rdfs:seeAlso
"https://www.scd.stfc.ac.uk/Pages/DL_POLY.aspx"^^
xs:anyURI;
  viso:is_tool_for_model viso:MM;
  viso:is_tool_for_model viso:MD;
  viso:has_feature viso:DOMAIN_DECOMPOSITION;
  viso:has_feature viso:DIRECT_COULOMB_SUM;
  viso:has_feature viso:SPME;
  viso:uses_language viso:FORTRAN90;
  viso:is_compatible_with :PLUMED;
  viso:is_compatible_with :OPENKIM;
  ...
```

Sub-domain specific ontologies

At a lower level with respect to VISO, we are developing in VIMMP **sub-domain ontologies** to be used to describe in detail the settings, input and output of the software tools.

These will allow:

1) **semantic interoperability** between the codes within an open simulation platform
2) **completeness and consistency check** of the input
3) **documentation of simulation results.**

The sub-domains are identified as classes of similar methods, such as particle-based simulations, continuum simulations, etc.

Assets with a similar purpose have been developed within the **NOMAD**[1] and **SimPhoNy**[2] projects. In the latter, **CUBA/CUDS** (Common Universal/Unified Data Structure and Basic Attributes) have been defined.

[1] NOMAD Metadata repository: https://gitlab.mpcdf.mpg.de/nomad-lab/nomad-meta-info

[2] SimPhoNy Metadata repository: https://github.com/simphony/simphony-common/tree/master/ontology

Significant collaboration and interactions acknowledged:

Aachen	–	Youness Bami, Andreas Fiseni, Georg Schmitz
Bologna	–	Emanuele Ghedini
Bremen	–	Welchy Leite Cavalcanti, Peter Schiffels
Cambridge	–	Gerhard Goldbeck, Alexandra Simperler
Trondheim	–	Jesper Friis

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 760907.

VIMMP
VIRTUAL MATERIALS MARKETPLACE

To subscribe for announcements, send an e-mail to vimmp-webinar@ifam.fraunhofer.de.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 760907.

VIMMP
VIRTUAL MATERIALS MARKETPLACE