

Knowledge Representation in EPNet

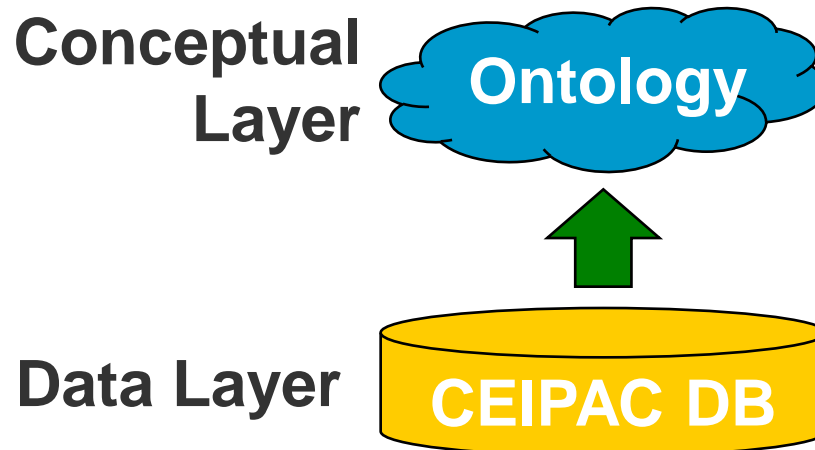
Alessandro Mosca and Guillem Rull



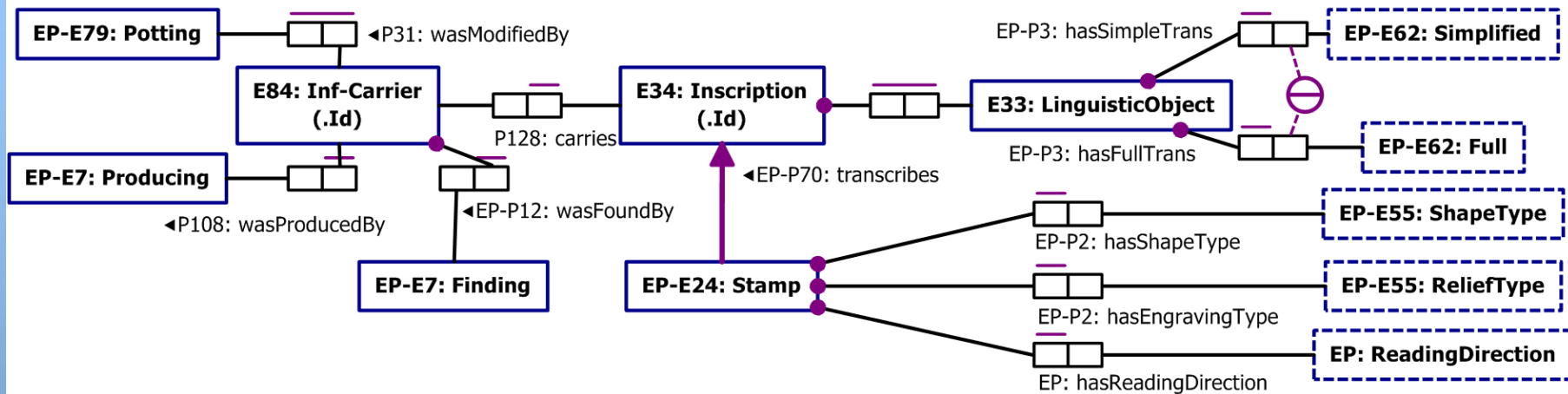
Conceptual Layer

■ Ontology ontop of the CEIPAC database.

- Vocabulary closer to the user than the one of the DB.
- Founded on logic, so allows reasoning.
- Allows answering queries that otherwise would yield no results.
 - Example: Asking for all the inscriptions in a case when the DB knows only about stamps and tituli.
 - Ontology needed to introduce concept Inscription, with Stamp and Titulus as particular cases (subconcepts).



Ontology



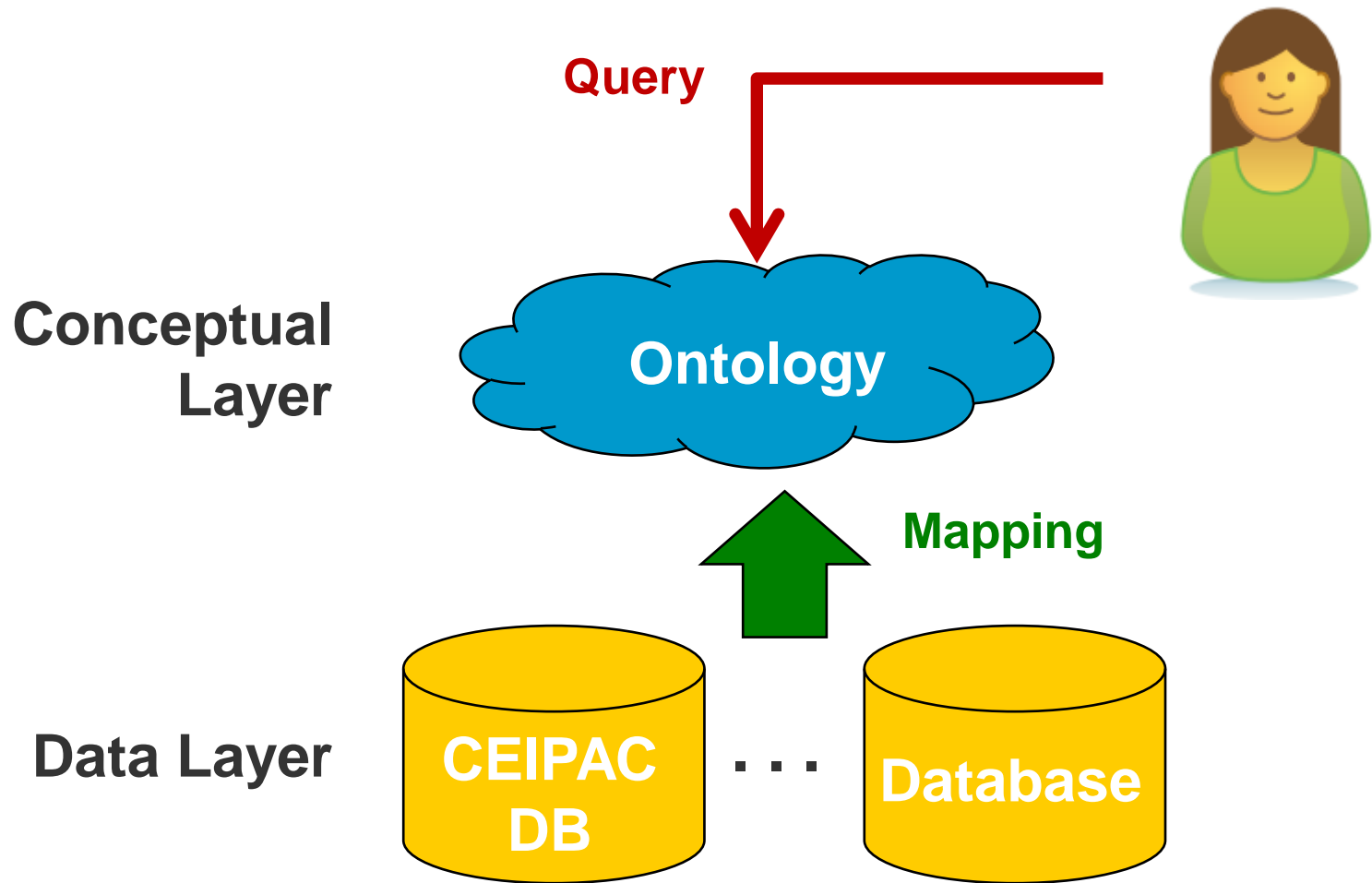
- Concepts:

- **Inf-Carrier** (i.e. an amphora), **Inscription**, **Stamp**, ...

- Properties:

- Inf-Carrier **carries** Inscription
- Inscription **isTranscribedBy** LinguisticObject
- LinguisticObject **hasSimplifiedTranscription** Text
- LinguisticObject **hasFullTranscription** Text

Architecture

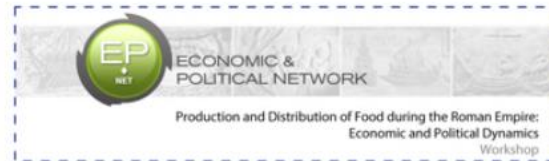


- The ontology also allows users to query **multiple heterogeneous data sources** using a common vocabulary.

Multiple Data Sources

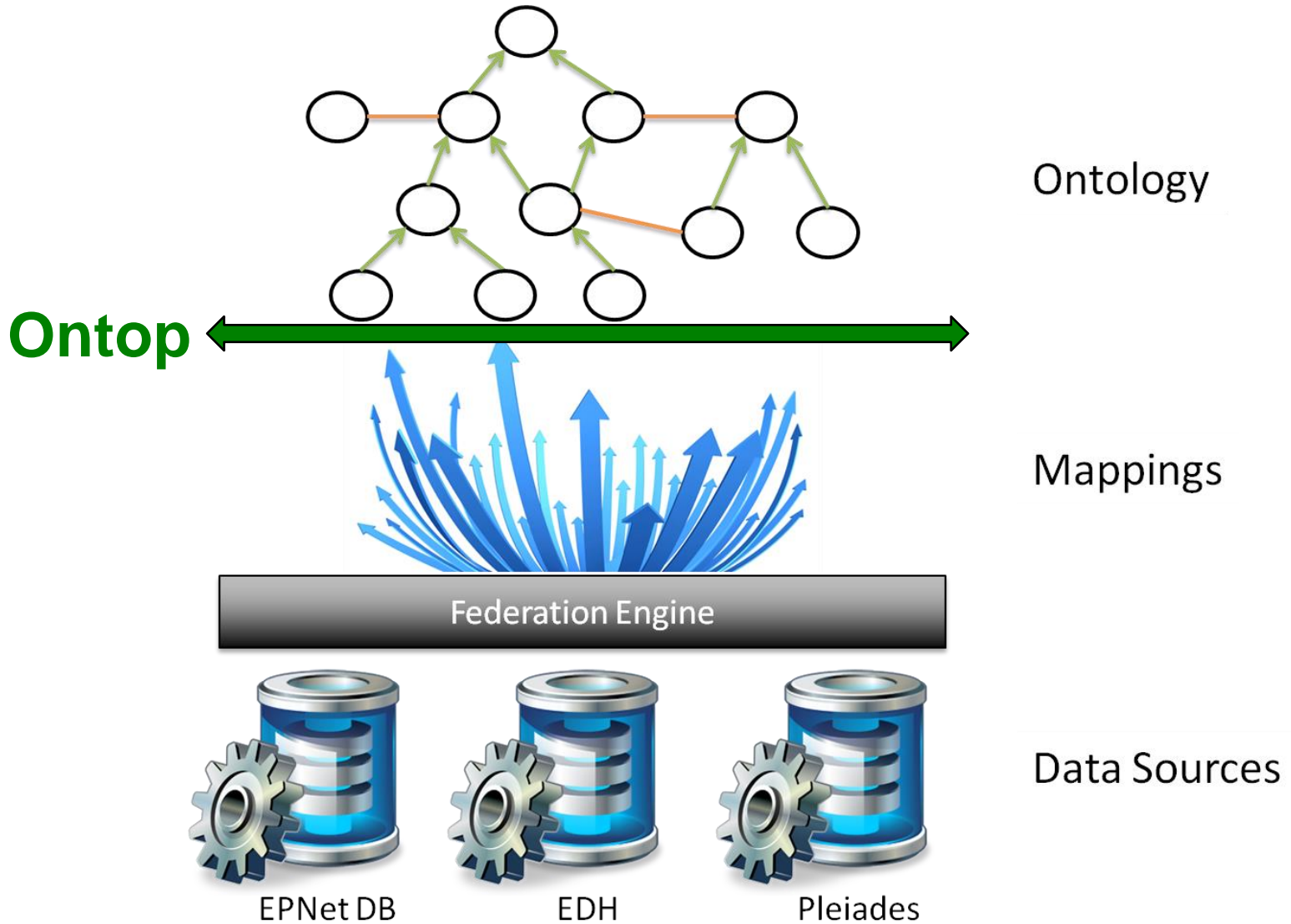


Roman Amphorae: a digital resource
University of Southampton, 2005 (updated 2014)



2015/11/30

Ontology-Based Data Integration using Ontop



Mapping between Data and Conceptual Layers

- **Set of rules** that tell how the data stored in the database(s) corresponds to the concepts and properties in the ontology.

`(:db1/{id}, type, :Amphora) ← SELECT amId AS id FROM AmphoraT`

`(:db1/{id}, :hasInscription, {insc}) ← SELECT amId AS id, insc FROM AmphoraT`

- From the point of view of the ontology **data looks like:**

`:Amphora-557 rdf:type :Amphora`

`:Amphora-557 :hasInscription "PNN"`

Querying the Data Layer

■ SPARQL language

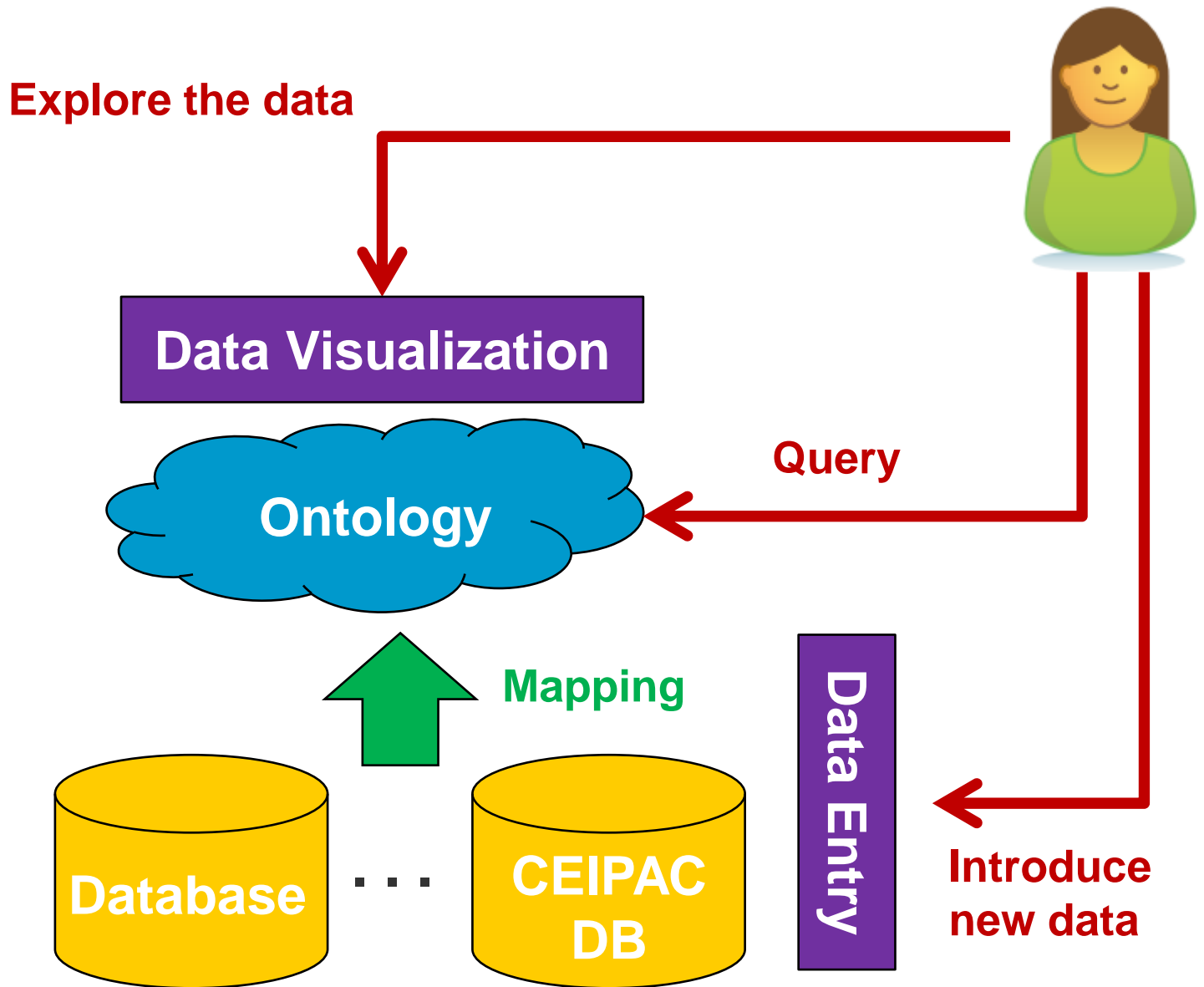
- Give me all amphorae, indicating their inscriptions (stamp or titulus), finding place (municipality and country) and geo-coordinates.

```
select ?transcription ?municipality ?latitude ?longitude
where {
  ?x rdf:type :Amphora .

  ?x :carries ?inscr .
  ?inscr rdf:type :Inscription .
  ?inscr :isTranscribedBy ?lingObj .
  ?lingObj :hasTranscription ?transcription .

  ?x :hasFindingPlace ?y .
  ?y :fallsWithin ?mun .
  ?mun rdf:type :Municipality .
  ?mun dcterms:title ?municipality .
  ?mun :hasLatitude ?latitude .
  ?mun :hasLongitude ?longitude .
}
order by ?municipality ?transcription
```


Graphical User Interfaces (GUIs)



GUIs vs SPARQL

■ GUIs

- Easy to use.
- Predefined set of queries (users can only ask those questions the interface allows them to).

■ SPARQL

- Requires some learning.
- User-defined queries (total freedom).
- The language allows for rather complex queries.

Demo

- Preliminary user interface for testing the OBDI functionalities in EPNet.

<http://136.243.8.213/epnet-pleiades-edh/>



← → ↻ 136.243.8.213/epnet-pleiades-edh/ ★ ☰

EP ECONOMIC & POLITICAL NETWORK

ontop
framework

SPARQL query:

```
PREFIX : <http://www.semanticweb.org/ontologies/2015/1/EPNet-ONTOP_Ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX dcterms: <http://purl.org/dc/terms/>
select *
where {
}
```

Thanks for your attention!