Knowledge Graph Engineering

Knowdive Research Group

September 5, 2023

Part 0
Course Organization

Objectives

- Learn how to produce quality and reusable data.
- Learn how to exploit reusable data for a (different) specific purpose.
- Learn what is a Knowledge Graph (KG).
- Learn a methodology to build quality and reusable KGs.
- Learn tool and instruments to implement the above methodology.

Prerequisites

- **Data management**: basic programming skills in python and/or java/javascript.
- **Databases modeling**: ER modeling, Ontology modeling if possible, Ontology definition desirable.
- **Attitude to teamwork**.

Course Modality - The Theory

The theory enables the practice

The theory lectures will be focused on:

- (First part of the course) Data Heterogeneity, Quality and Reuse.
- (Second part of the course) The iTelos methodology for KG building.

Course Modality - The Practice

The course practical activities apply the iTelos methodology in real-world case studies.

- The practical activities are scheduled in parallel with the theory lectures.
- The students (grouped in teams) will have to conduct a complete KG generation project (focused on real case studies assigned by tutors).

Course Modality - Following the course

- Theory and practice will go on in parallel.
- The theoretical lectures will describe the problems to solve, and the solutions proposed by the iTelos methodology;
- those will be then immediately applied in practice over the assigned projects.

It is strongly suggested:

- The presence in the classroom for the theoretical lectures and their following discussions.
- Strong cooperation between the team members is required to carry on the project's development along the course.

Exam Modality - Intermediate evaluations

- After the completion of each iTelos phase (both concerning theory and practice) the students will have to provide an **intermediate report** of the work done so far.
- The intermediate evaluation will allow the tutors to lead the teams towards the right direction by correcting possible errors during the methodology implementation.

Exam Modality - Final evaluation

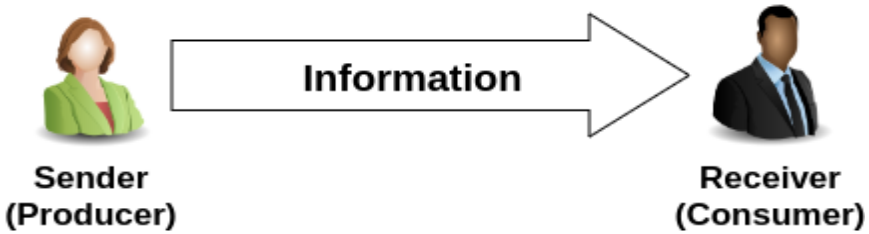
- The final exam will consist of a presentation of the KGE projects developed along the course and finalized achieving the output required by the initial purpose.
- Additional questions will be asked by the tutors over the both the course theory and practice.
- The course final grades will be composed by the grades obtained for each intermediate evaluations plus the final presentations grade

Part 1
The Reuse Problem

Part 1.1
Information & information reuse

Information

The information is stimuli (i.e., electromagnetic waves), created by a sender, **that has meaning in some context** for its receiver.



- The information is represented, using its raw form, by the **data**. The data, represented and managed in different ways, transports the information within a **communication**, from sender to receiver.
- Notice how a communication can be the creation of data (Producer) to be exploited by any kind of data service (Consumer).

Information Reuse

- The information in a communication is not always new, most of the time instead is **reused** from previous communication.
- As a consequence, the **reuse of information, and thus data**, is crucial in a communication between sender(producer) and receiver(consumer).

What does it means reuse of data ?

The data reuse is defined over three components:

1. Data representation
2. Data reuse processes
3. Data architecture for reuse

Part 1.2
Data representation

Data Representation

- The data needs to represent different aspects that the information has to express in a communication.
- Reusable data is present, and available, in a (data) **world that is, apparently, disordered**.
  - "disorder": high level of **heterogeneity**, low quality, requiring **huge amount of pre-processing**.
- The data appears in multiple forms, highlighting what is called **data heterogeneity**.

Data Representation - Heterogeneity

- The general meaning of *Heterogeneity* is the *"quality or state of consisting of dissimilar or diverse elements"*[1].
- Heterogeneity is the key distinguishing feature of life: there will never be two identical moments, two identical places, two identical individuals, or two datasets!
- *Data Heterogeneity*, therefore, is the principle bottleneck in achieving reuse and integration data.

[1] https://www.merriam-webster.com/dictionary/heterogeneity

Data Representation - Heterogeneity

Data heterogeneity is defined over four encapsulated sub-layers of heterogeneity:

1. Source heterogeneity
2. Format heterogeneity
3. Structure heterogeneity
4. Meaning heterogeneity

Data Representation - Source Heterogeneity

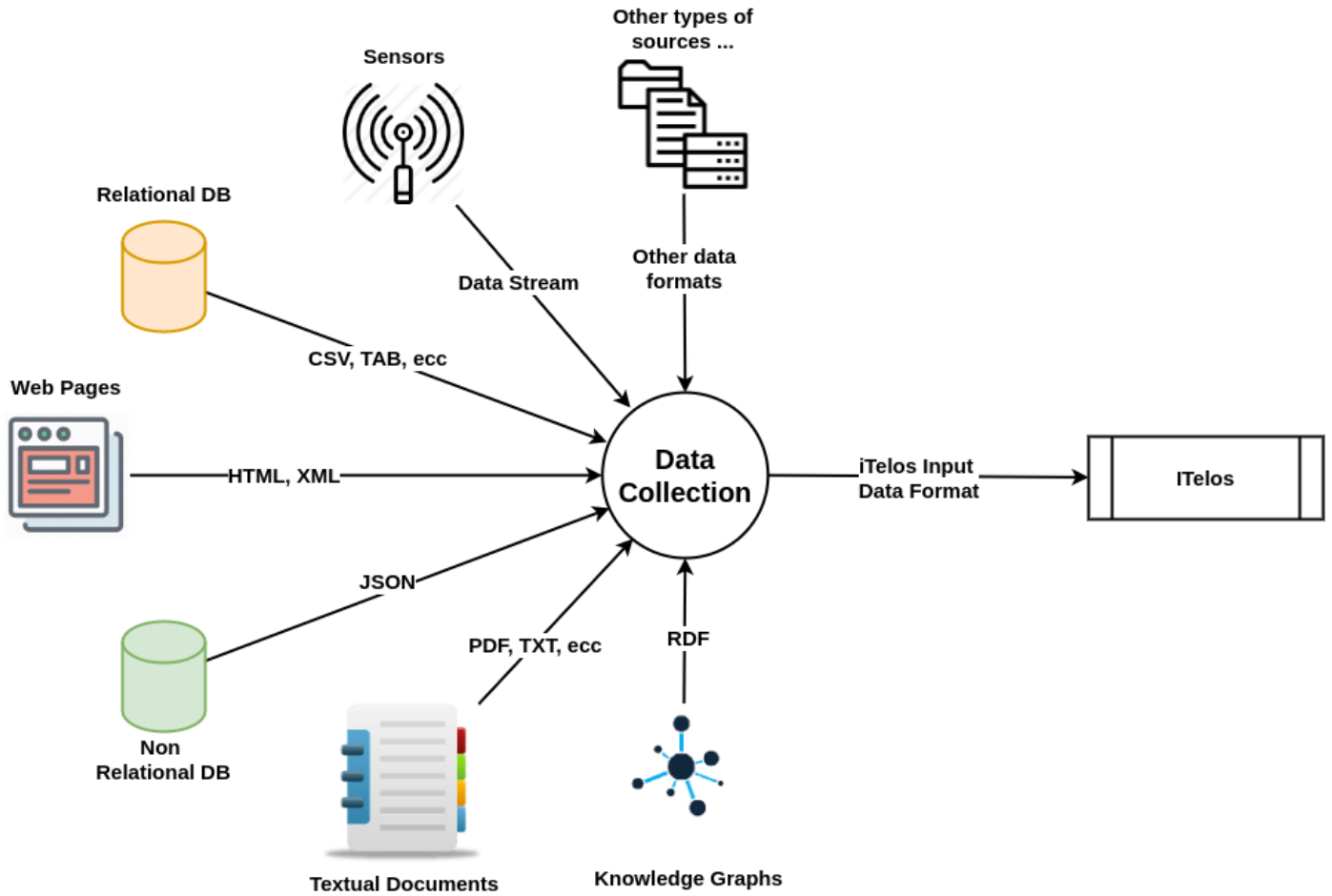
- Information can be transmitted through different modes, for instance, via:
  - Visual Mode.
  - Linguistic Mode.
  - Aural Mode.
  - Gestural Mode, ... etc.!
- Within each such mode, there can be several possible information sources, e.g.,:
  - Visual: Art, Photos, Videos, etc.
  - Linguistic: Written text in different languages.
  - Aural: Music, Speech, etc.
  - .... etc.!

Data Representation - Source Heterogeneity

- Source Heterogeneity refers to the *diverse possible sources of information* that can be employed to differently record information about the same *target reality*.
- For example, information about the same car can be differently recorded via:
  - Datasets recording different properties of the car.
  - Written textual description of the car in different languages.
  - Photos of the car from different angles.
  - Videos of the car from different angles.
  - A speech about the car.
  - ... etc.!

Source heterogeneity

- The heterogeneity at source layer



Data Representation - Format Heterogeneity

- For each source of information, there can be possibly many types of data *formats* which can be used to encode information about a target reality.
- For example, following are some formats which employ different syntax to encode information:
  - Images: JPEG, PNG, TIFF, BMP, SVG, etc.
  - Videos: WebM, MKV, FLV, etc.
  - Text: Doc, PDF, RTF, TXT, etc.
  - Datasets: JOSN, CSV, RDF, etc.
  - .... etc.!

Data Representation - Format Heterogeneity

- Even assuming the same source, Format Heterogeneity refers to the *diverse possible data formats* that can be employed to differently encode information about the same *target reality*.
- For example, information about the same car can be differently encoded via:
  - Datasets recording different properties of the car in CSV or JSON or RDF.
  - Written textual description of the car in different languages in PDF or DOC or TXT.
  - Photos of the car from different angles in JPEG or PNG.
  - Videos of the car from different angles in FLV or MKV.
  - ... etc.!

Data Representation - Structure Heterogeneity

1. Even assuming the same source and format, heterogeneity appears over the structure of the information within the data, that we can call **Structure Heterogeneity**.

2. Structure Heterogeneity is conventionally understood as the existence of variance in the representation and description of the same target reality, e.g., of the car, when modeled through different properties by different sources.

3. Structure heterogeneity (but in general all layers of heterogeneity) appears at **three different levels** within the data:
   - Language
   - Knowledge
   - Data

Data Representation - Structure Heterogeneity (SH) - Language

SH in Language (LH) refers to the different levels of abstraction in the concepts employed to describe the same target reality in a language. For example:

Car (LH)				
Nameplate	schema: speed	schema: fuelCapacity	schema: fuelType	schema: modelDate
FP372MK	150	62	Petrol	2020-11-25

Vettura (LH)		
Targa	Velocità	Tipo di corpo
FP372MK	158	Coupé

Vehicle (LH)			
vso:VIN	vso:feature	vso:modelDate	vso:speed
FP372MK	Armrest	2020-11-25	155.0

Data Representation - Structure Heterogeneity (SH) - Knowledge

SH in Knowledge (KH) refers to the different (set of) properties employed to describe the conceptualization of the same target reality. For example:

Car				
Nameplate	schema: speed	schema: fuelCapacity	schema: fuelType	schema: modelDate
FP372MK	150	62	Petrol	2020-11-25

KH

Vettura		
Targa	Velocità	Tipo di corpo
FP372MK	158	Coupé

KH

Vehicle			
vso:VIN	vso:feature	vso:modelDate	vso:speed
FP372MK	Armrest	2020-11-25	155.0

KH

Data Representation - Structure Heterogeneity (SH) - Data

SH in Data (DH) refers to the different (set of) data values (belonging to different data types) employed to describe the conceptualization of the same target reality. For example:

Car				
Nameplate	schema: speed	schema: fuelCapacity	schema: fuelType	schema: modelDate
FP372MK	150	62	Petrol	2020-11-25

> DH

Vettura		
Targa	Velocità	Tipo di corpo
FP372MK	158	Coupé

> DH

Vehicle			
vso:VIN	vso:feature	vso:modelDate	vso:speed
FP372MK	Armrest	2020-11-25	155.0

> DH

Data Representation - Meaning heterogeneity

- Even fixing a source of information from which data is collected and represented through a specific data formats, as well as adopting clear data structures, a final layer of heterogeneity has to be considered.
- **Meaning Heterogeneity**, is defined over the values of the information properties which can be used to identify a real world entity, thus distinguishing one entity from one another.

Data Representation - Meaning heterogeneity

Example: consider the Car entity represented in two different datasets A, and B.

Car in dataset A:

- Vehicle-ID: 1234
- Manufacturer: "Renault"
- Engine-type: "Electric engine"
- Fuel-type: "Electricity"

Car in dataset B:

- Vehicle-ID: ABCD
- Manufacturer: "Renault"
- Engine-type: "Electric engine"
- Fuel-type: "Electricity"

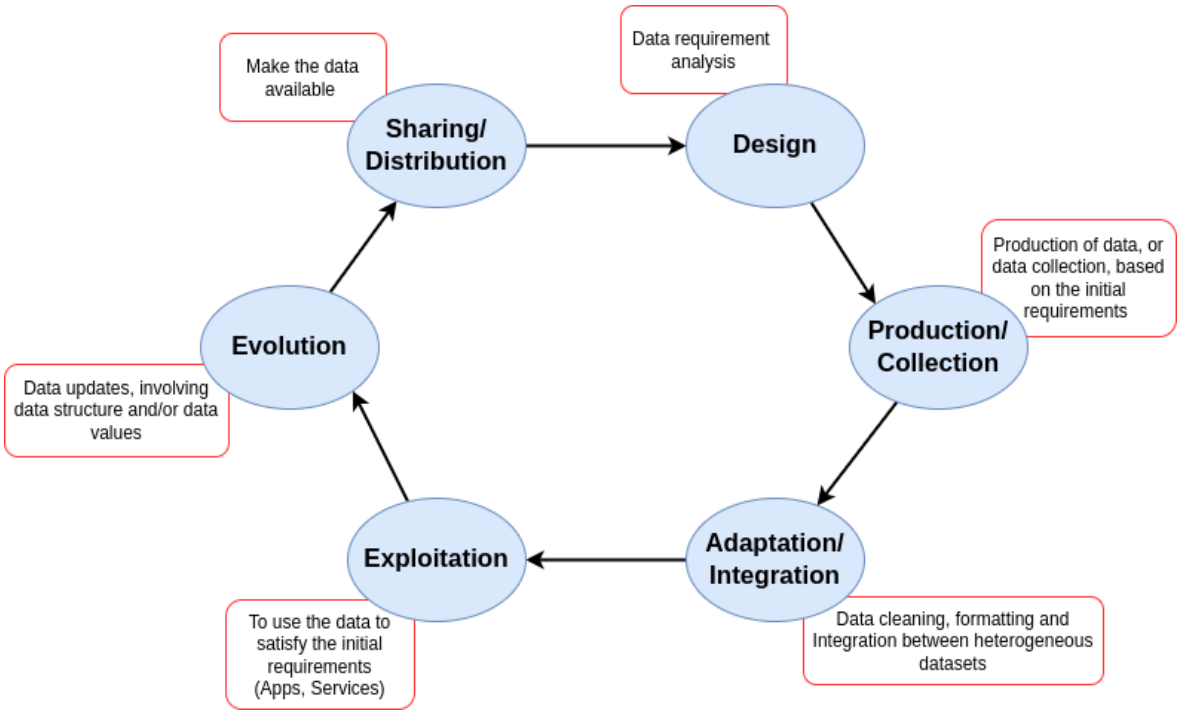
From the same source, we have two datasets in the same format, using the same structure of information. Nevertheless ..

- how can we know if the two car are the same entity or different ones ?
- is the identifier in dataset A equivalent to the identifier in dataset B ?
- the "Manufacturer" term in datasets A has the same meaning of "Manufacturer" in dataset B ?

Part 1.3
Data reuse processes

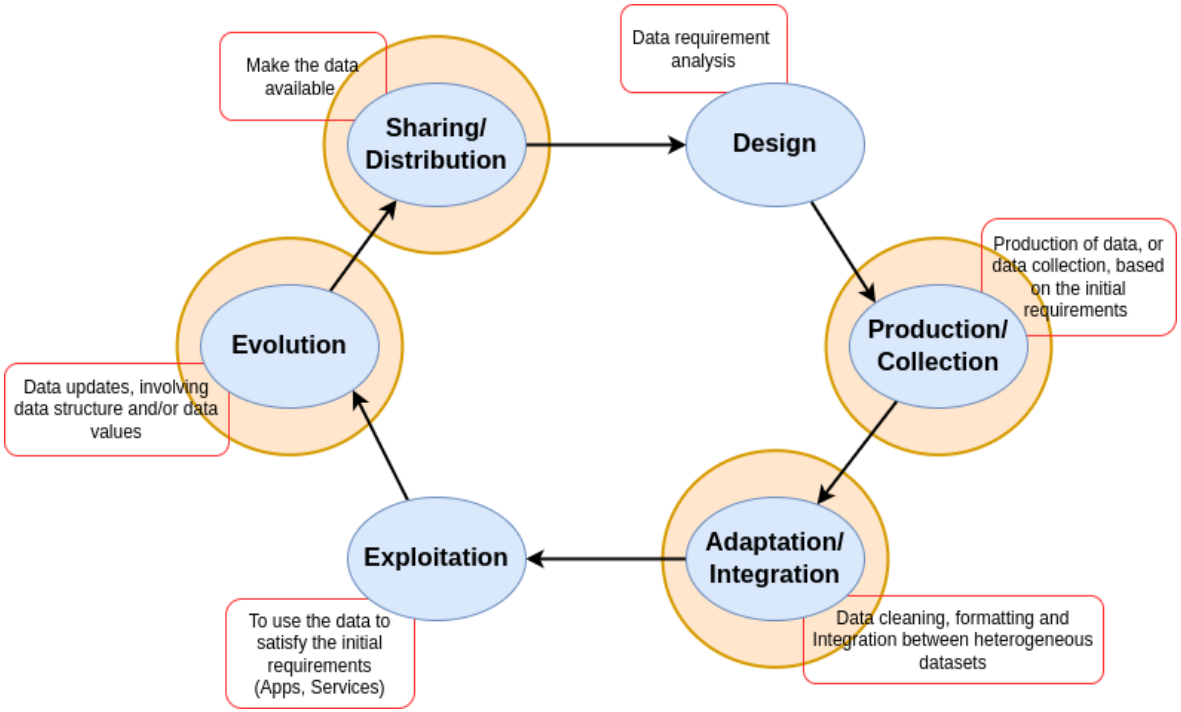
Data Reuse Processes

- The reuse of data, is not only a matter of representing data, it involves also **the processes** required **to get, exploit**, and **make reusable** such data.
- To understand the role of such processes in data reuse, we can see them into **the data life cycle**.



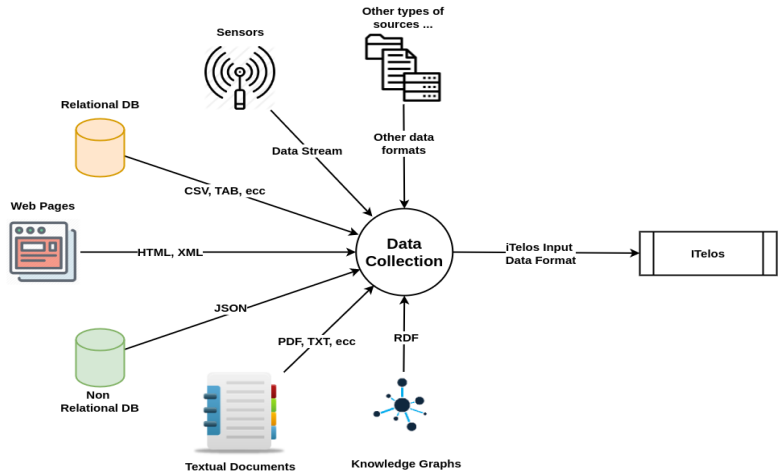
Data Reuse Processes

- Data life cycle activities most involved in data reuse.



Data Reuse Processes - Data Collection

- The data collection activity requires processes for the extraction (scraping) of data from data sources.
- Such processes need to be defined and implemented considering the **source heterogeneity**.
  - Potentially each data source requires a dedicated implementation of data collection processes, thus increasing the effort to be paid for the whole data life cycle.



Data Reuse Processes - Data Production

- The production of data does not affect the **reuse of existing data**.
- Nevertheless, it plays a very crucial role in **the reuse of new data**, being th activity responsible of the creation of data which can be potentially reused.
- Data production processes that do not consider reusability and interoperability of data, increase the overall cost of the data life cycle.

Data Reuse Processes - Data Adaptation/Integration

- **Data adaptation**: activity which aims at cleaning and formatting (format heterogeneity) the data to be exploited for a specific purpose.
- **Data integration**: activity which aims at integrate together different datasets to obtain a merged information resource (structure heterogeneity), able to satisfy a specific purpose.

The KGE course is strongly focused on Data Integration processes.

Data Reuse Processes - Data Adaptation/Integration

- The adaptation and integration processes are fundamental in the reuse of data, mainly for two reasons:

1. (input side) The efficiency of such processes has a strong impacts over the data life cycle.
   - **cleaning**: how much the a reusable datasets can be cleaned out from noise, respect to a specific purpose to be satisfied ?
   - **formatting**: which, and how many, standards the process is able to apply to the dataset to be formatted ?
   - **integration**: how much the integration process is able to deal with Data Heterogeneity ?
2. (output side) The way the data are cleaned, formatted and integrated, strongly affects their future reusability.

Data Reuse Processes - Data Evolution

- **Is a data able to scale up ?**
  - How much effort is required **to extend the data produced/collected and adapted/integrated**, in order to satisfy new feature, respect the data initial purpose ?
    - evolution at schema level (schema update);
    - evolution at data level (data values update). For example, data expiration
- Low quality data evolution processes can increase the cost to be paid in the data life cycle.

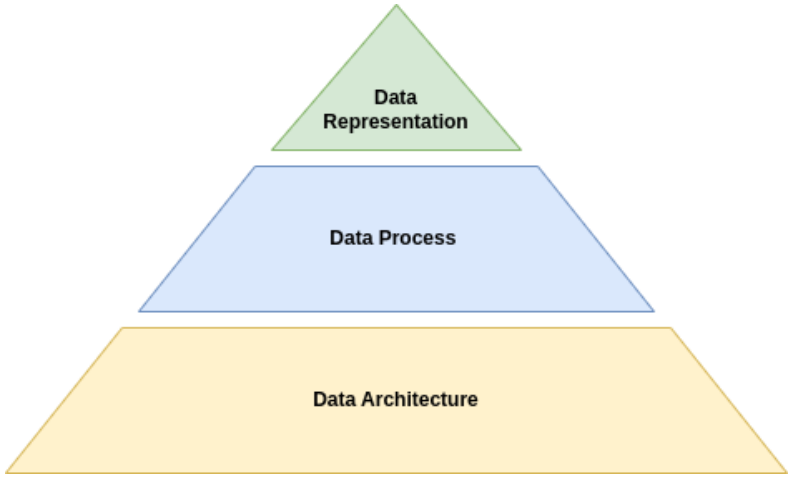
Data Reuse Processes - Data Sharing

- As already anticipated by discussing about previous data reuse processes, **the reuse is not only a matter of getting existing data in input**.
- The reuse of data strongly involves the process for **data sharing** (or data distribution).
- Low quality data sharing processes introduce difficulties for the retrieval of the data, thus **limiting its potential future reuse**.

Part 1.4
Data architecture for reuse

Data Architecture for Reuse

- The data needs to be represented properly to handle its heterogeneity, and
- process are required to implement the reuse of data.
- Moreover, to fully address the data reuse problem, we need to consider the environment in which such a data can be properly represents, and the reuse processes correctly supported.
- Such environment is defined by the **architecture** (or infrastructure) **enabling the reuse of data**.



Data Architecture for Reuse

The fundamental requirements required for such architectures are:

- **Data collection support**: components and services for the upload of reusable data (Fundamental to support the data collection processes).
- **Data store support**: storage components and services.
- **Data elaboration support**: component and services supporting the adaptation, integration and evolution processes.
- **Data distribution support**: component and services supporting the data sharing processes.

The lack of data architectures where the above requirements are not considered, and/or not well composed together, limits the possibilities of data reuse.

Part 2
State of the Art

Part 2.1
Data representation SoA

Data Representation SoA

- To reuse data we need to know **which data, and which types of data, are available**, so that we can identify the most suitable resources for a specific purpose.
- To this end, let's discover
  - the different types of data available;
  - how they are represented, and,
  - which are the existing best practices to enhance data quality and interoperability.

Data Representation SoA - Types of data

Data can be recognized and classified in many different ways. In this course, to describe the available types of data, we focus on 2 key dimensions;

- Cross-sectional data and Time-series data
- Domain data and Person-centric data

Data Representation SoA - Types of data

- **Cross-sectional data**: it carries information about a single moment in time. It doesn't consider the evolution of the data along the time.

  "*is a type of data collected by observing many subjects (such as individuals, firms, countries, or regions) at the one point or period of time.*"[2]

- **Time-series data**: it carries information about multiple moments in time. It describe the properties of one, or more, entities considering their evolution in time.
  - A concrete example of time-series data, is the data collected periodically by using sensors like, gps and accelerometer.

[2] https://en.wikipedia.org/wiki/Cross-sectional_data

Data Representation SoA - Types of data

- **Domain data**: this kind of data carries information about a specific domain of interest, by describing the entities composing it.
- **Person-centric data**: this kind of data carries information about the human behavior, thus describing a person, and her/his point of view within a specific domain of interest (or context).

The domain data provides the background data space where the person-centric data can be contextualized. In other words the **environment** (domain data) where one, or more, **subjects** (person-centric data) act.

Data Representation SoA - Existing languages and formats

- The data classified as above, is represented in several different languages and formats (as already described discussing the data heterogeneity).
- Sometimes the data is represented by using **tabular formats**, like:
  - JSON
  - CSV, TSV
  - Excels spreadsheet
- Other times the data is represented by using **graph-based formats**, like:
  - XML
  - RDF-OWL
- For this course, the graph-based data representation is particularly relevant, because that is the way in which **Knowledge Graphs** (KG) are represented [3]

[3] We will see how to solve the reuse problem by exploiting the Knowledge Graphs

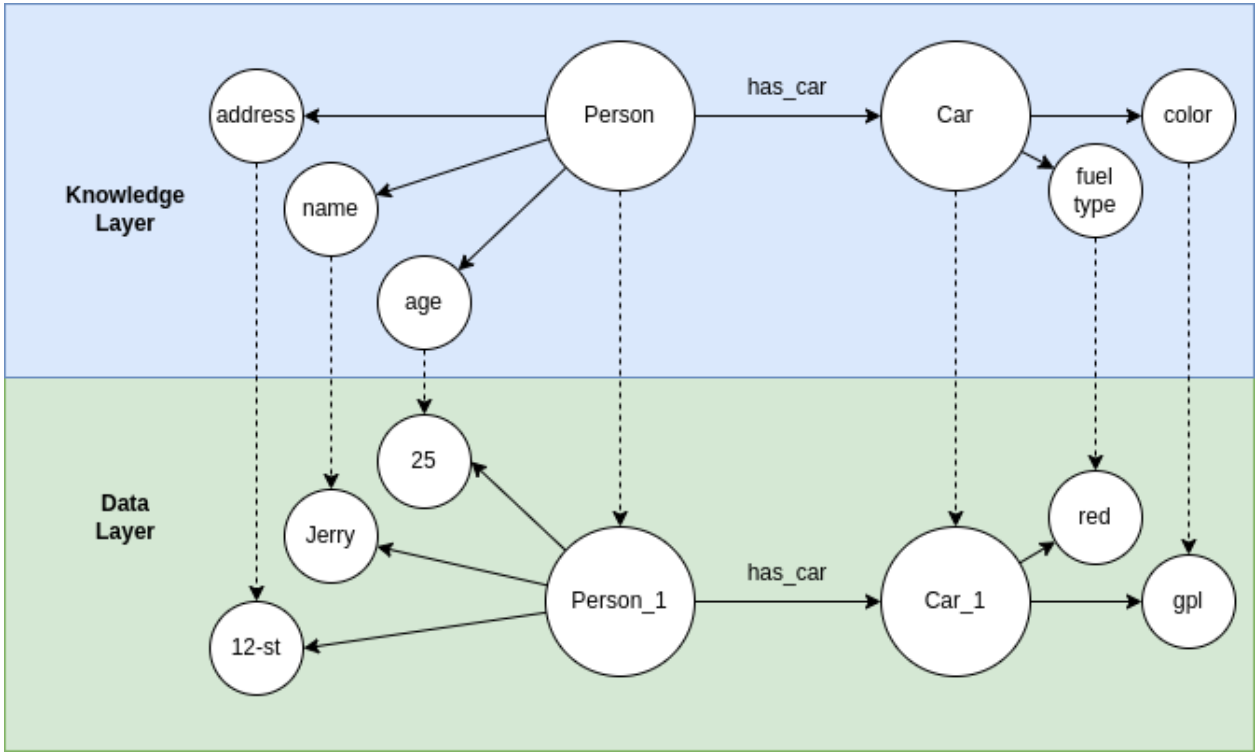
Knowledge Graph definition

A Knowledge Graph K, can be defined as follows:

$$KG = (E, D, R, A)$$

Where:

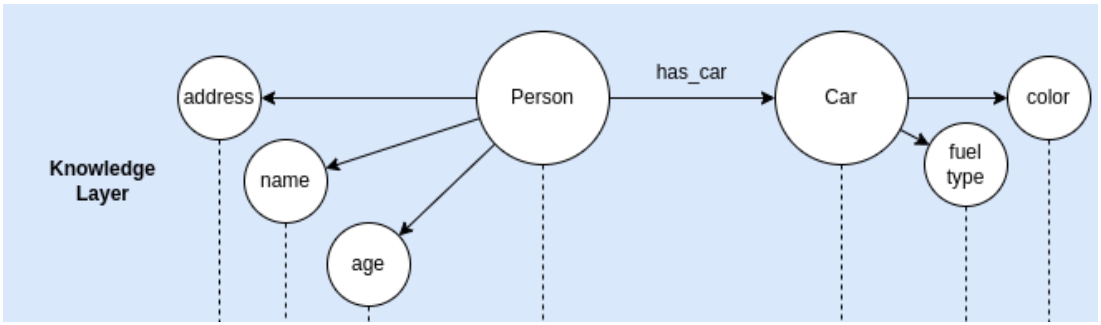
- E: is the set of real-world objects types, called *Entity Types* (or ETypes).
- D: is the set of real-world objects representations, called *Entities*. The Entities are ETypes instantiation.
- R is the set of properties used to denote the ETypes. The elements of R, can be properties related to a single EType, called *data properties*, or properties used to define relations among different ETypes, *called object properties*.
- A: is the set of property values denoting the attributes of the Entities. Each attribute, associated to one and only one property, instantiates the relative data/object property.



- E = {Person, Car}
- D = {Person_1, Car_1}
- R = {address, name, age, color, fuel type, has_car}
- A = {12-st, Jerry, 25, red, gpl}

Knowledge Layer

- The KG's Knowledge Layer is composed by the elements of E (ETypes) plus the element of R (properties definition).
- It defines the KG's structure (or schema).
- It is usually defined using an ontology modeled to represent the information to be maintained in the KG.



Ontology

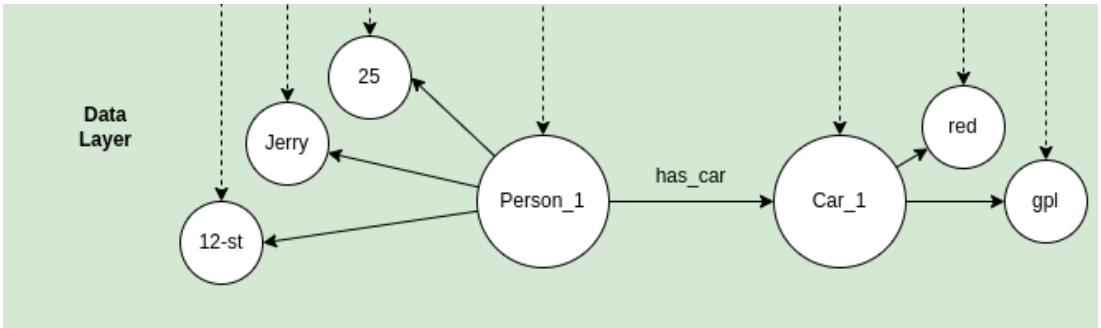
- "An ontology is a formal, explicit specification of a shared conceptualization"
  -by Gruber (1993) and modified by Studer et. al (1998)

- Ontologies are used to capture knowledge about some domain of interest. An ontology describes the concepts in the domain and also the relationships that hold between those concepts

- Ontologies are crucial for attributing semantics to Knowledge Graphs (KGs) which model ground-truth

Data layer

- The KG's Data Layer is composed by the elements of D (Entities) plus the element of A (attributes definition).
- It contains the data values instantiating the KG's structure.



KG-based Apps - examples

- Google Knowledge Panel

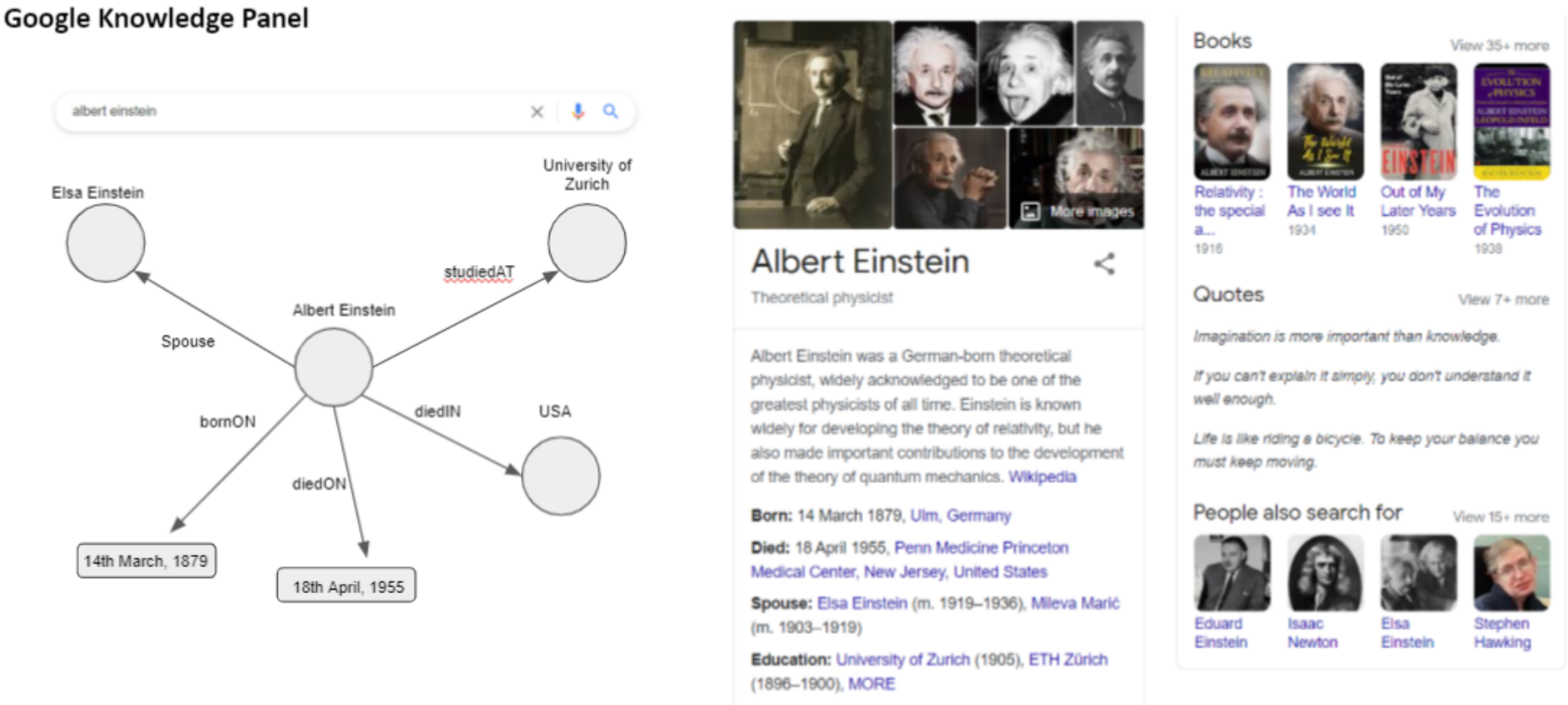


Figure: Mohit M., A guide to Knowledge Graphs, Aug 30, 2021

KG-based Apps - examples

- InteropEHRate EU project

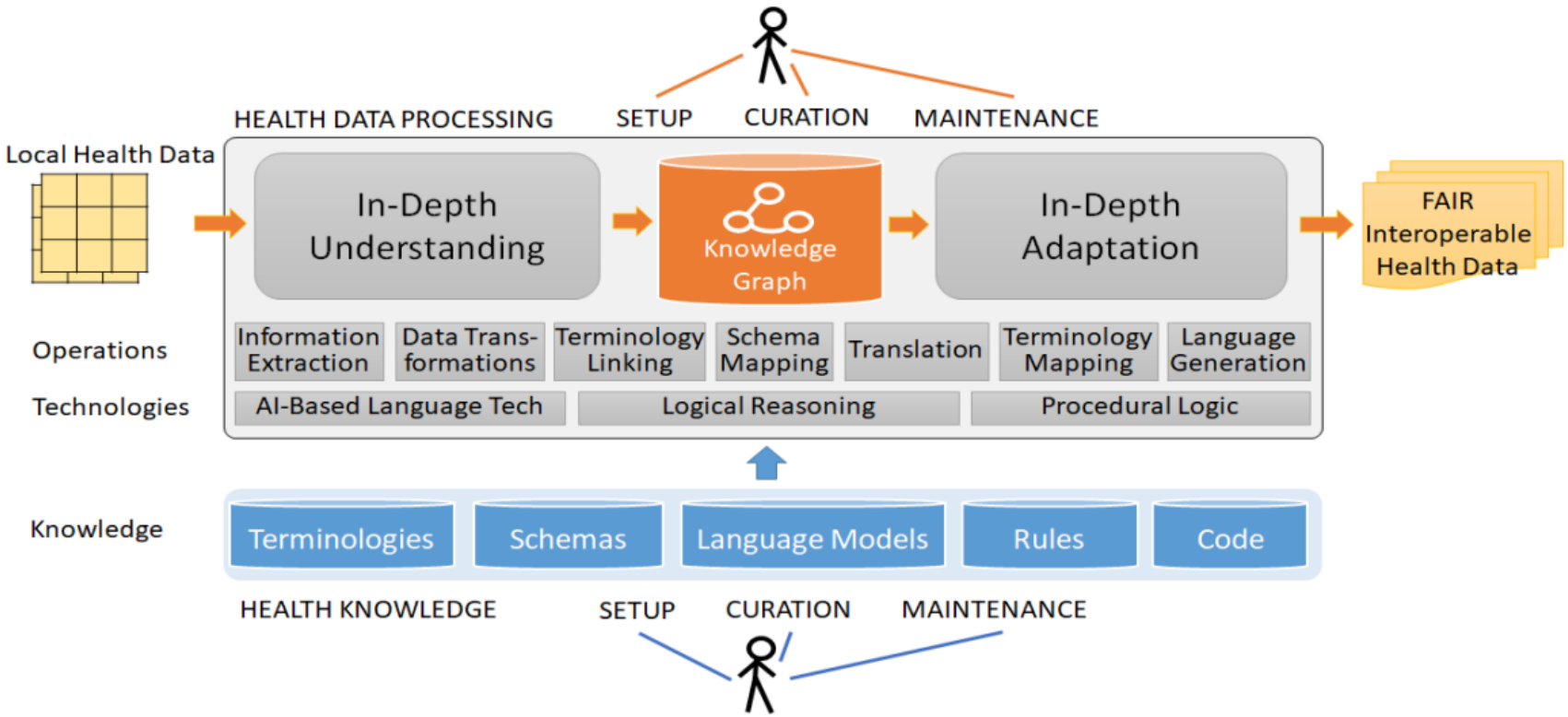


Figure 1: High-level architecture of the InteropEHRate Health Services and the way they are overseen by a human data manager.

KG-based Apps - examples

Many domain specific KGs have been produced supporting tasks like:

- Data Governance
- Automated Fraud Detection
- Knowledge Management
- Insider Trading
- Health Data Interoperability

KG-based Apps - examples

While there are several small-sized and domain-specific KGs, on the other hand, we also have many huge-sized and domain-agnostic KG that contains facts of all types and forms.

- **DBpedia**: is a crowd-sourced community-based effort to extract structured content from the information present in various Wikimedia projects.
- **Freebase**: a massive, collaboratively edited database of cross-linked data. Touted as "an openly shared database of the world's knowledge". It was bought by Google and used to power its own KG. In 2015, it was finally discontinued.
- **OpenCyc**: is a gateway to the full power of Cyc, one of the world's most complete general knowledge base and commonsense reasoning engines.
- **Wikidata**: is a free, collaborative, multilingual database, collecting structured data to provide support for Wikimedia projects.
- **YAGO**: huge semantic knowledge base, derived from Wikipedia, WordNet, and GeoNames.

Data Representation SoA - Quality and Interoperability

- To enhance the quality and interoperability of data, some criteria and best proctices have been already defined.
  - Open (5⋆) data
  - FAIR data

Open (5⋆) data

"The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data."[4]

- ⋆ Make your stuff available on the Web (whatever format) under an open license
- ⋆ ⋆ Make it available as structured data (e.g., Excel)
- ⋆ ⋆ ⋆ Use non-proprietary open format (e.g., CSV instead of Excel)
- ⋆ ⋆ ⋆ ⋆ Use URIs to denote things, so that people can point at your stuff
- ⋆ ⋆ ⋆ ⋆ ⋆ Link your data to other data to provide context

[4] Tim Berners-Lee, Linked data-design issues, https://www.w3.org/DesignIssues/LinkedData.html, 2006.

FAIR data

Findability	**F1.** (Meta)data are assigned a globally unique and persistent identifier
	F2. Data are described with rich metadata (defined by R1 below)
	F3. Metadata clearly and explicitly include the identifier of the data it describes
	F4. (Meta)data are registered or indexed in a searchable resource
Accessibility	**A1.** (Meta)data are retrievable by their identifier using a standardised communications protocol **A1.1.** The protocol is open, free, and universally implementable **A1.2.** The protocol allows for an authentication and authorisation procedure, where necessary
	A2. Metadata is accessible, even when the data are no longer available
Interoperability	**I1.** (Meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation
	I2. (Meta)data use vocabularies that follow FAIR principles
	I3. (Meta)data include qualified references to other (meta)data
Reusability	**R1.** Metadata is richly described with a plurality of accurate and relevant attributes **R1.1.** (Meta)data are released with a clear and accessible data usage license **R1.2.** (Meta)data are associated with detailed provenance **R1.3.** (Meta)data meet domain-relevant community standards

Figure: FAIR data principles [5]

[5] https://www.go-fair.org/fair-principles/

FAIR is not Open

- "FAIR data and open data are two distinct concepts which are however coming closer and closer. (Mons, etal., 2017) distinguish the concept of FAIR from the concept of open, saying: In the envisioned Internet of FAIR Data and Services, the degree to which any piece of data is available, or even advertised as being available (via its metadata) **is entirely at the discretion of the data owner**."
- "... moreover, the Council of the European Union concluded that "**as open as possible, as closed as necessary**" is the underlying principle for optimal reuse of research data."

Cost of non-FAIR data

The European Commission report [6] (March 2018) indicates that:
"*the annual cost of not having FAIR research data costs the European economy at least €10.2bn every year*"

[6] European Commission. "Cost of not having FAIR research data-Cost-Benefit analysis for FAIR research data." (2018)

Cost of non-FAIR data - Research activities involved

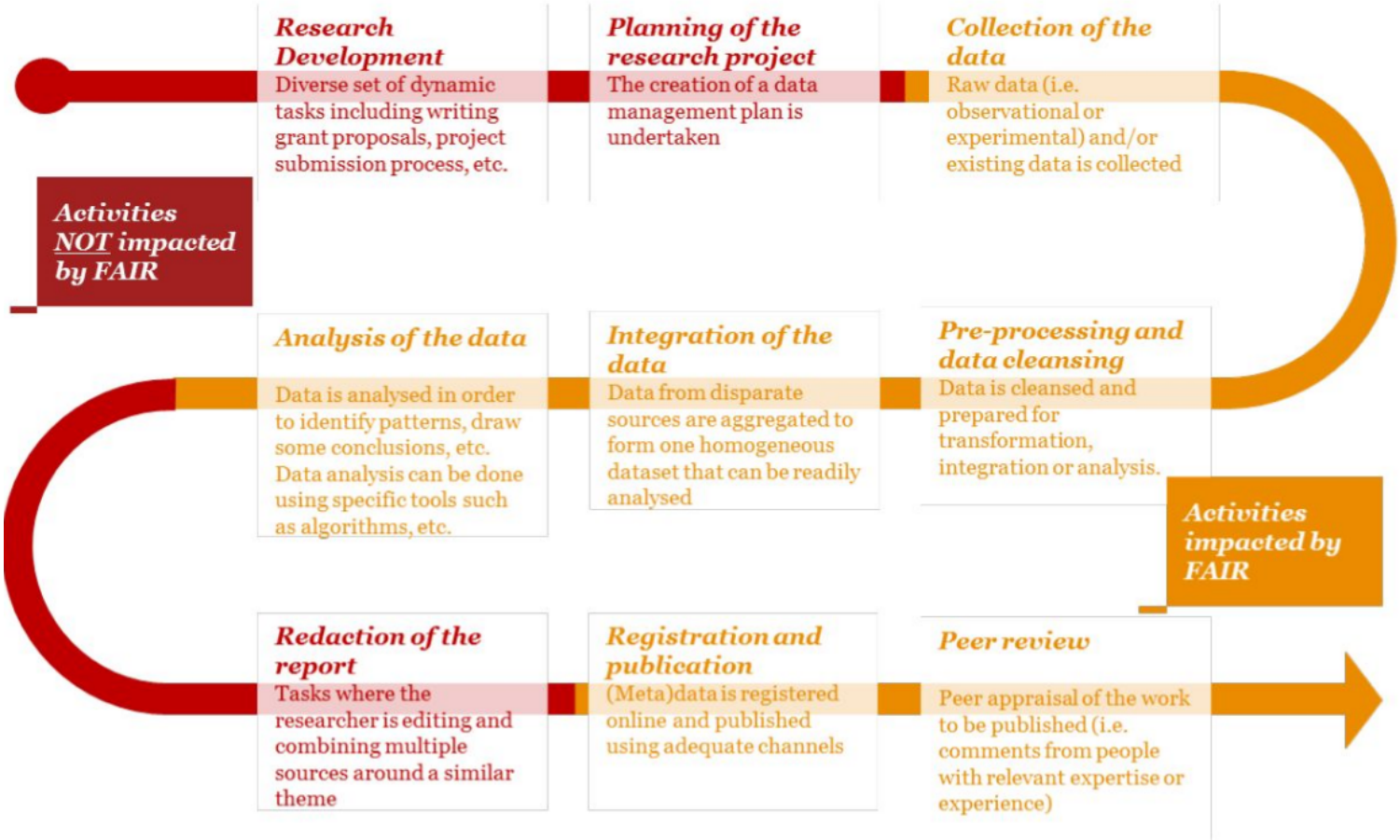


Figure: Cost of not having FAIR research data-Cost-Benefit analysis for FAIR research data." (2018)

Part 2.2
Reusable Resources

Data Representation SoA - Reusable resources

- Which data resources are already available to be reused and where we can find them ?
- Depending by the information carried, we can find three different types of reusable data:
  - Linguistic
  - Knowledge
  - Data values

Open data Catalogs

- Where are the reusable resources we need to build KGs ?
- Several projects and open data portal already exist which allow to retrieve useful resources.
- Often such resources are accessible through **Catalogs**. They are open portals collecting information about several resources (i.e. datasets, schemas, ontologies, ...).
- The catalogs doesn't collect the real resources, but instead the **metadata** describing such resources. (Catalogs are supported by backhand repositories)
- More metadata are associated to a resource, more detailed it is on the catalog, thus by consequence, it will be more findable and **reusable**.

Linguistic Resources

A linguistic resource is a dataset which provides data about languages (e.g., meanings, relations between words, ...).
There are two types of mono/multi-lingual resources: (i) online dictionaries and (ii) Wordnet like resources. Wordnets much more useful in data integration as they connect meanings of words in a LKG.
Check the licence (lots of options).

Example

- Global Wordnet Association
- WordNet
- Open Multilingual WordNet
- Datascientia/UKC (forthcoming)

Linguistic Resource Repositories

Figure: Global WordNet Association[7]

Linguistic Resource Repositories

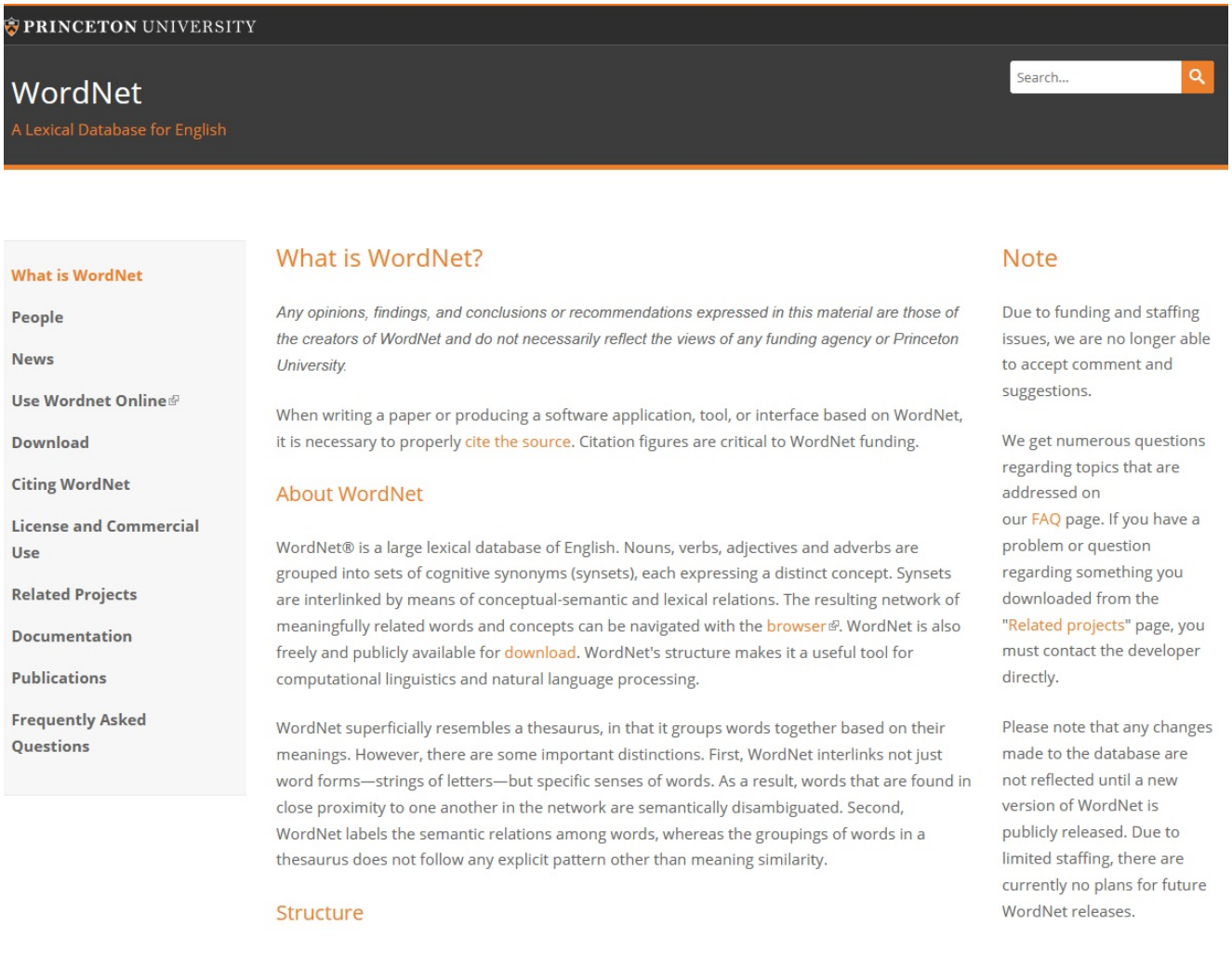


Figure: WordNet Home[8]

Linguistic Resource Repositories

Open Multilingual Wordnet

This page provides access to open wordnets in a variety of languages, all linked to the Princeton Wordnet of English (PWN). The goal is to make it easy to use wordnets in multiple languages. The individual wordnets have been made by many different projects and vary greatly in size and accuracy. We have (i) extracted and normalized the data, (ii) linked it to Princeton WordNet 3.0 and (iii) put it in one place. The Open Multilingual Wordnet and its components are open: they can be freely used, modified, and shared by anyone for any purpose. There is a fuller list of wordnets at the Global Wordnet Association's Wordnets in the World page.

If you use these wordnets, please cite the original projects who created them (linked in Table 1), if you got value from this aggregation/normalization, please cite Bond and Paik (2012).

You can access the wordnets through the (python) Natural Language Tool-Kit wordnet interface (NLTK).

We have an extended version with automatically extracted data for over a 150 languages from Wiktionary and the Unicode Common Locale Data Repository (Bond and Foster, 2013).

Documentation, News and Updates

Search

We have a simple search interface (search the extended wordnet). It uses the SQL database originally developed by the Japanese Wordnet.

34 Open Wordnets Merged

Wordnet	Lang	Synsets	Words	Senses	Core	Licence	Data	Citation
Albanet	als	4,675	5,988	9,599	31%	CC BY 3.0	als.zip (+xml)	cite:als; (.bib)
Arabic WordNet (AWN v2)	arb	9,916	17,785	37,335	47%	CC BY SA 3.0	arb.zip (+xml)	cite:arb; (.bib)
BulTreeBank Wordnet (BTB-WN)	bul	4,959	6,720	8,936	99%	CC BY 3.0	bul.zip (+xml)	cite:bul; (.bib)
Chinese Open Wordnet	cmn	42,312	61,533	79,809	100%	wordnet	cmn.zip (+xml)	cite:cmn; (.bib)

Figure: Open Multilingual WordNet Home[9]

Linguistic Resource Repositories

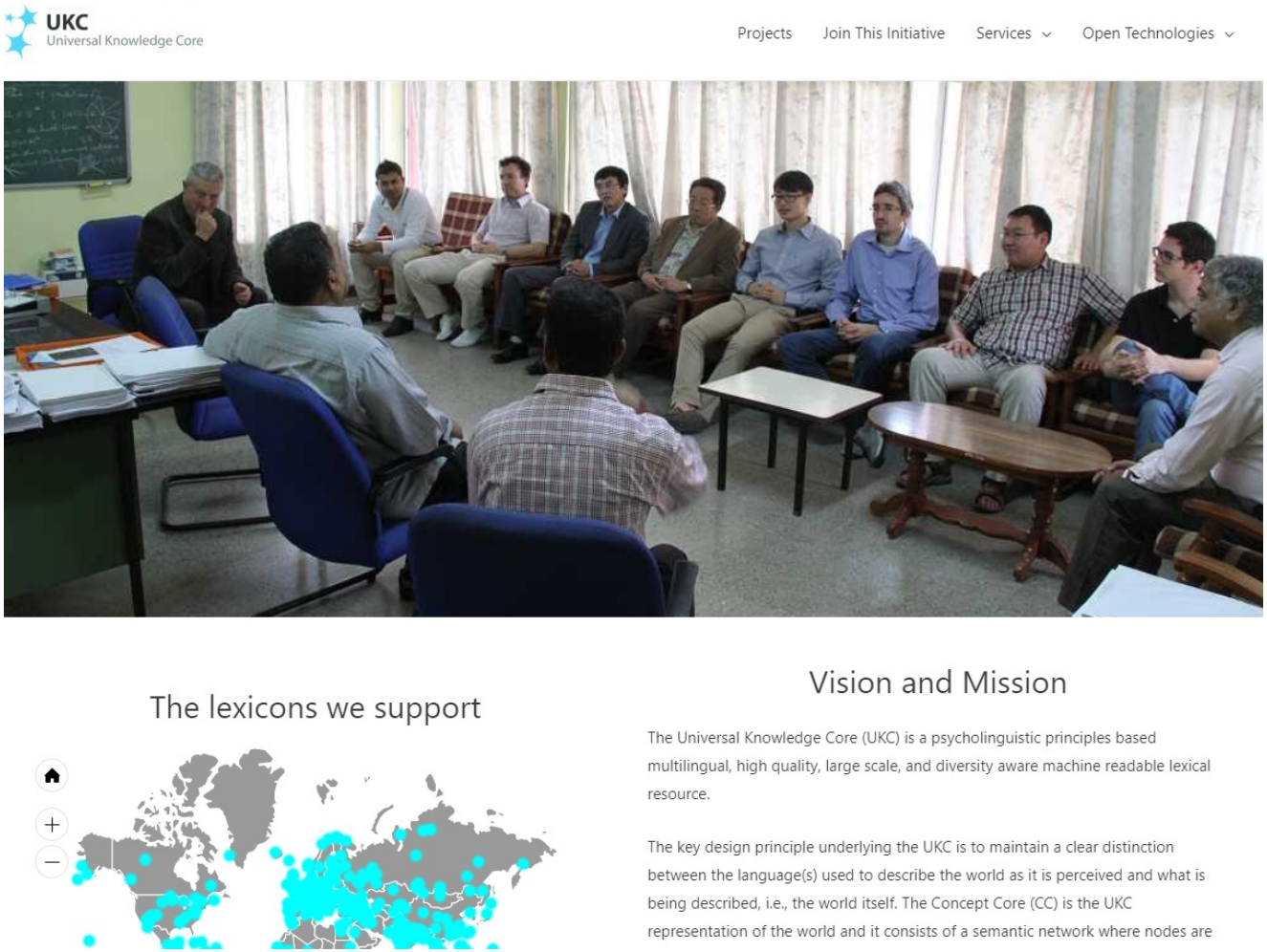


Figure: UKC Home[10]

Knowledge Resources

A Knowledge resource is a dataset which consists of a KB encoding information about schemas (etypes and properties).
KBs of high quality are usually called ontologies. We call them teleologies (meaning by this, ontologies with metadata which empower their practical use in knowledge and data integration).

Example

- LOV/LOV4IoT
- Schema.org
- DBpedia (schema only)
- Datascientia/liveschema (forthcoming)

Knowledge Resource Repositories

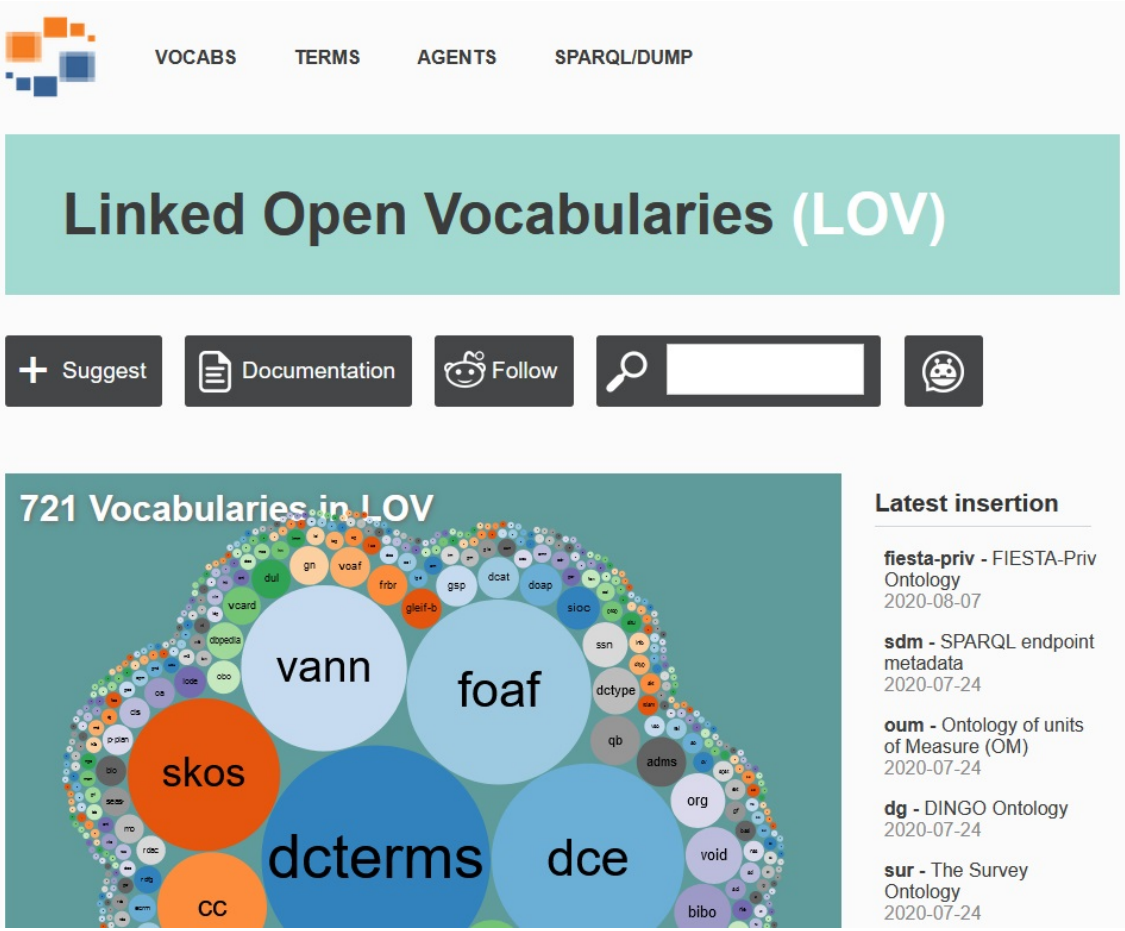


Figure: Linked Open Vocabulary[11]

Knowledge Resource Repositories

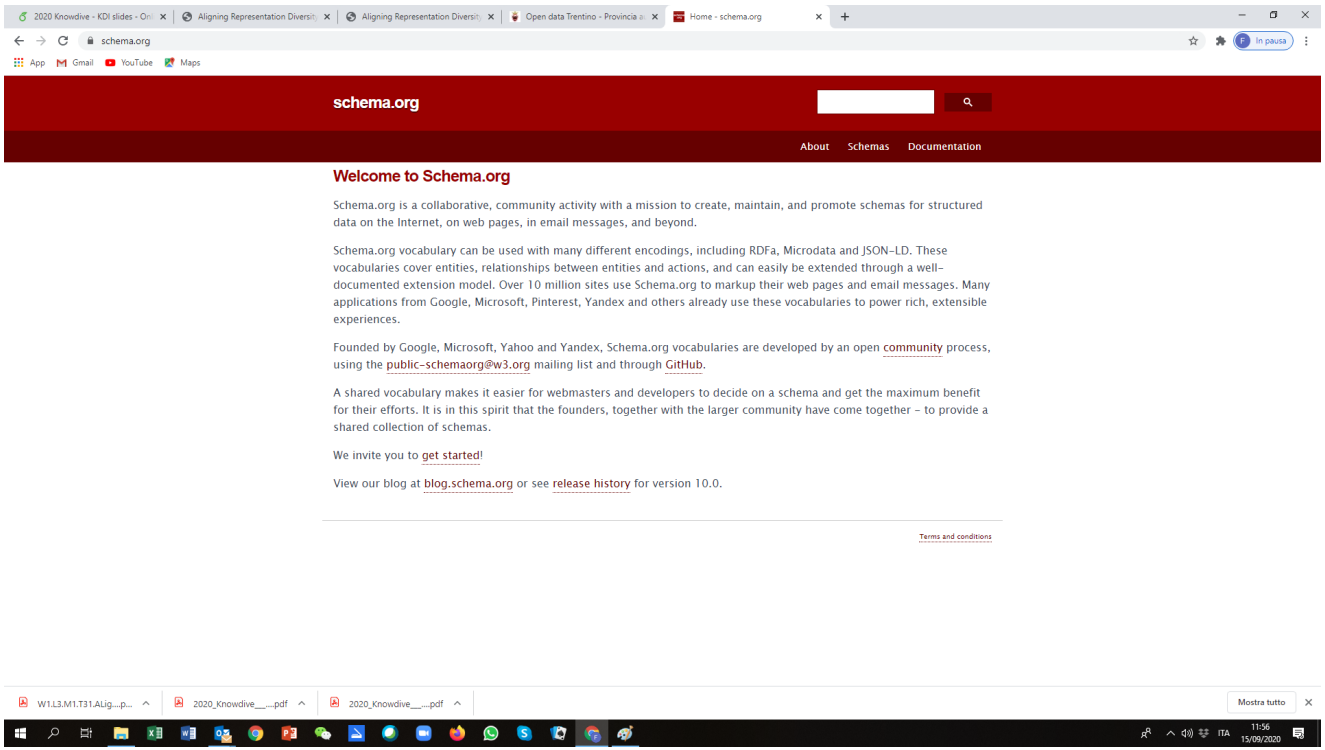


Figure: Schema.org[12]

[12] http://www.schema.org/

Knowledge Resource Repositories

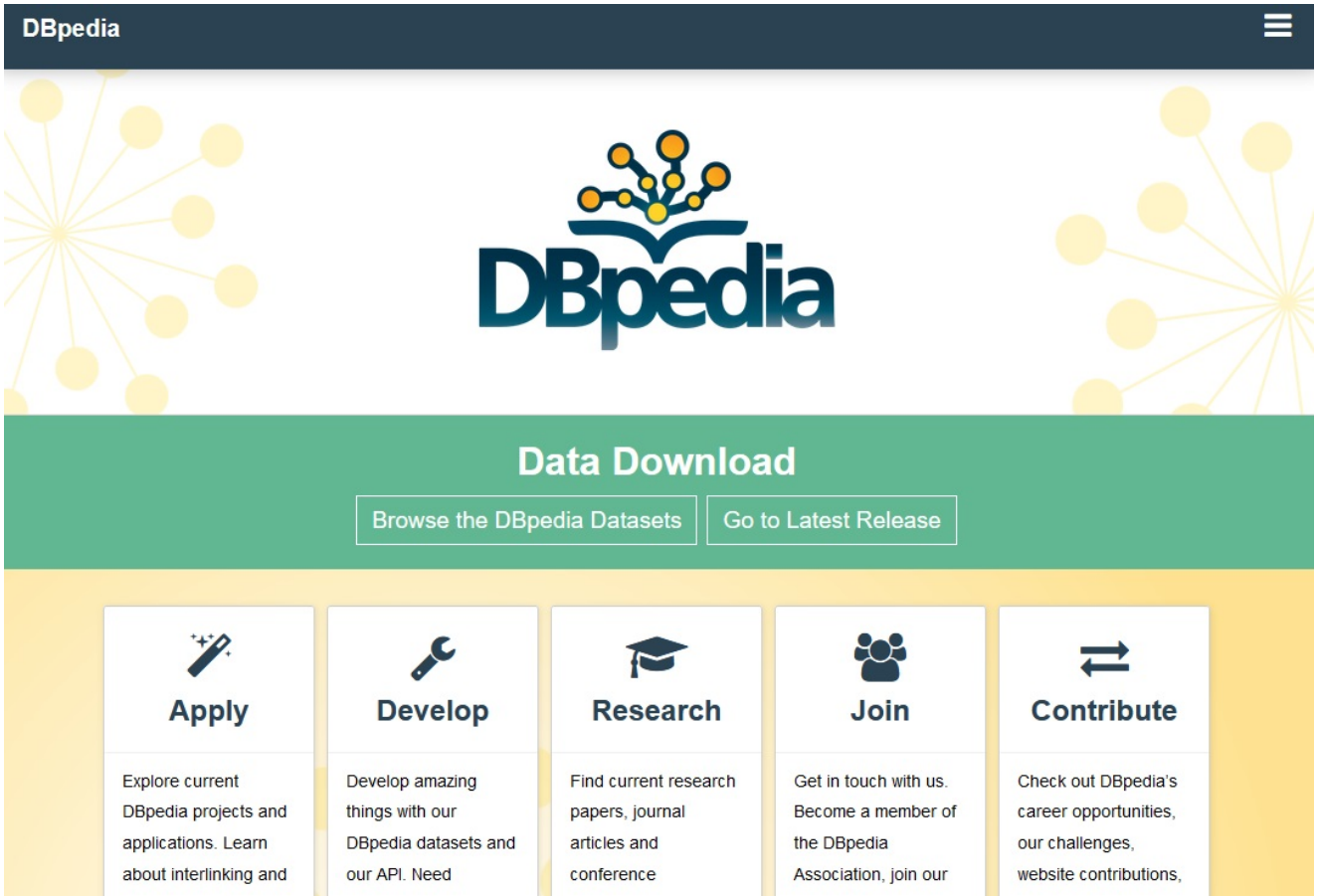


Figure: DBpedia Home[13]

Knowledge Resource Repositories

Figure: DataScientia Catalogs[14]

Data Resources

A data resource is a dataset which consists of data in some format (tabular, unstructured, entities and property values).

Example

- UK Open Data
- National Bureau of Statistics, China
- data.org
- Opendata Trentino (see, among others, Unitn Open Data)
- Geonames
- Open Street Map
- DBPedia
- Data Hub

Data Resource Repositories

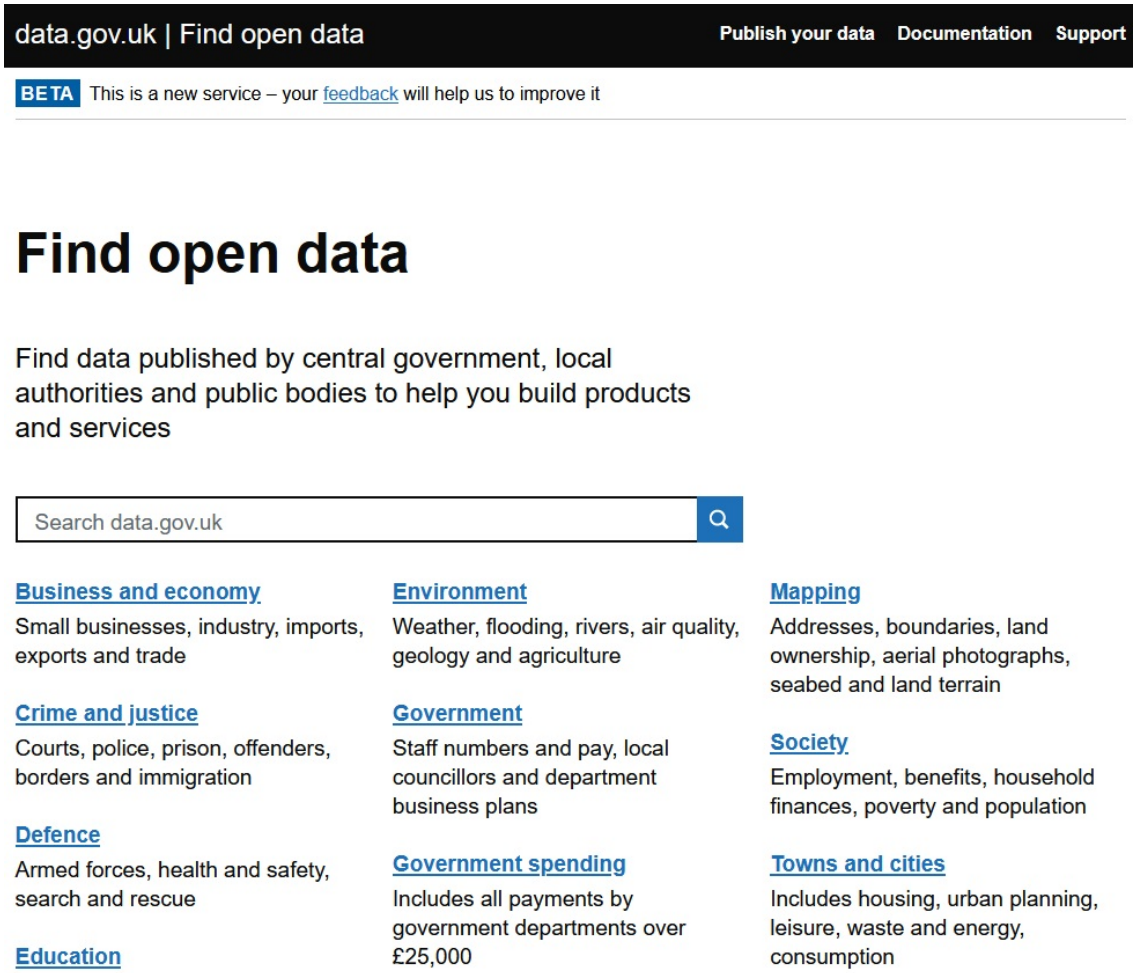


Figure: Open Data UK[15]

Data Resource Repositories

Figure: National Bureau of Statistics, China[16]

Data Resource Repositories

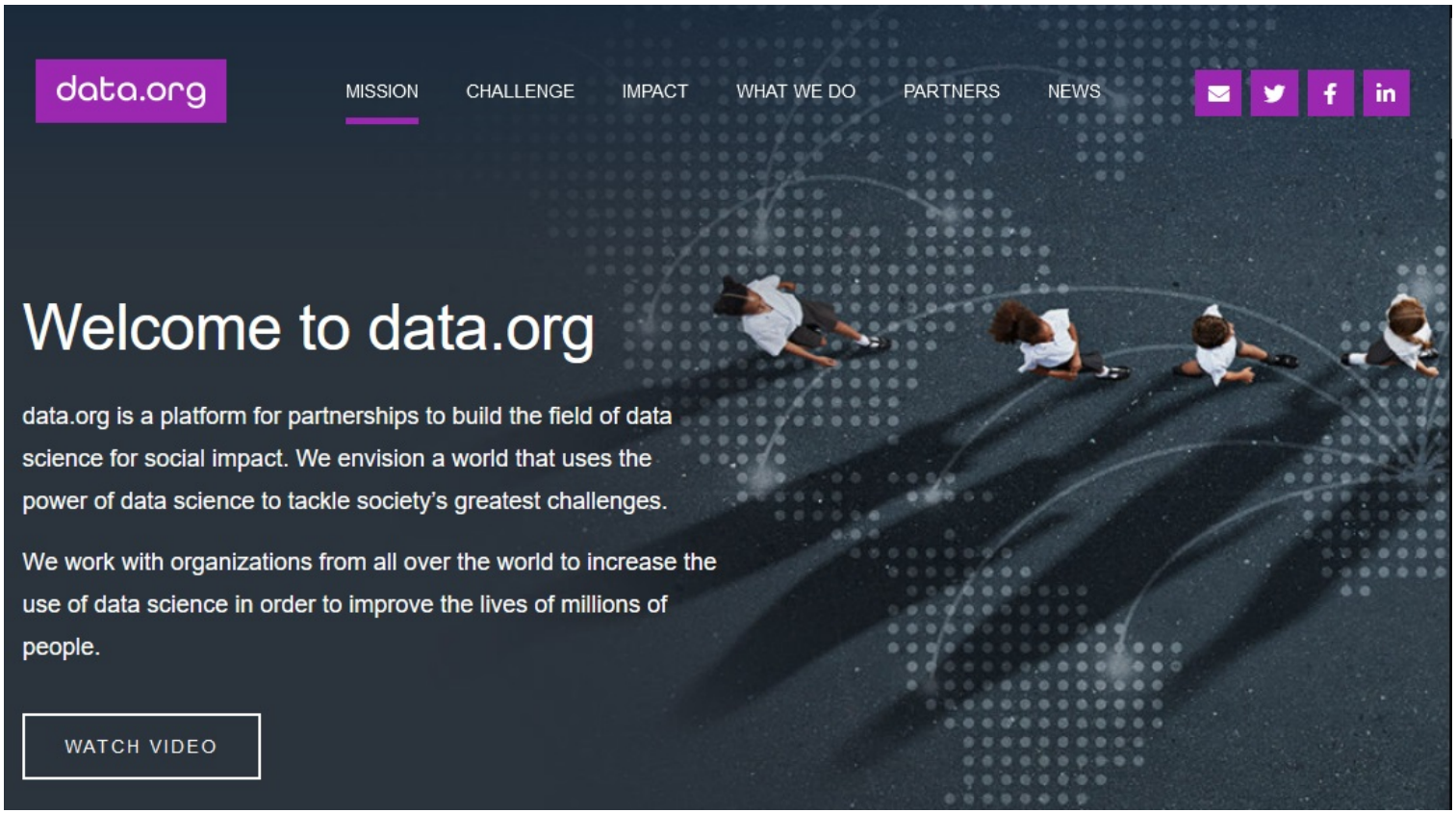


Figure: data.org[17]

Data Resource Repositories

Figure: Open Data Trentino[18]

Data Resource Repositories

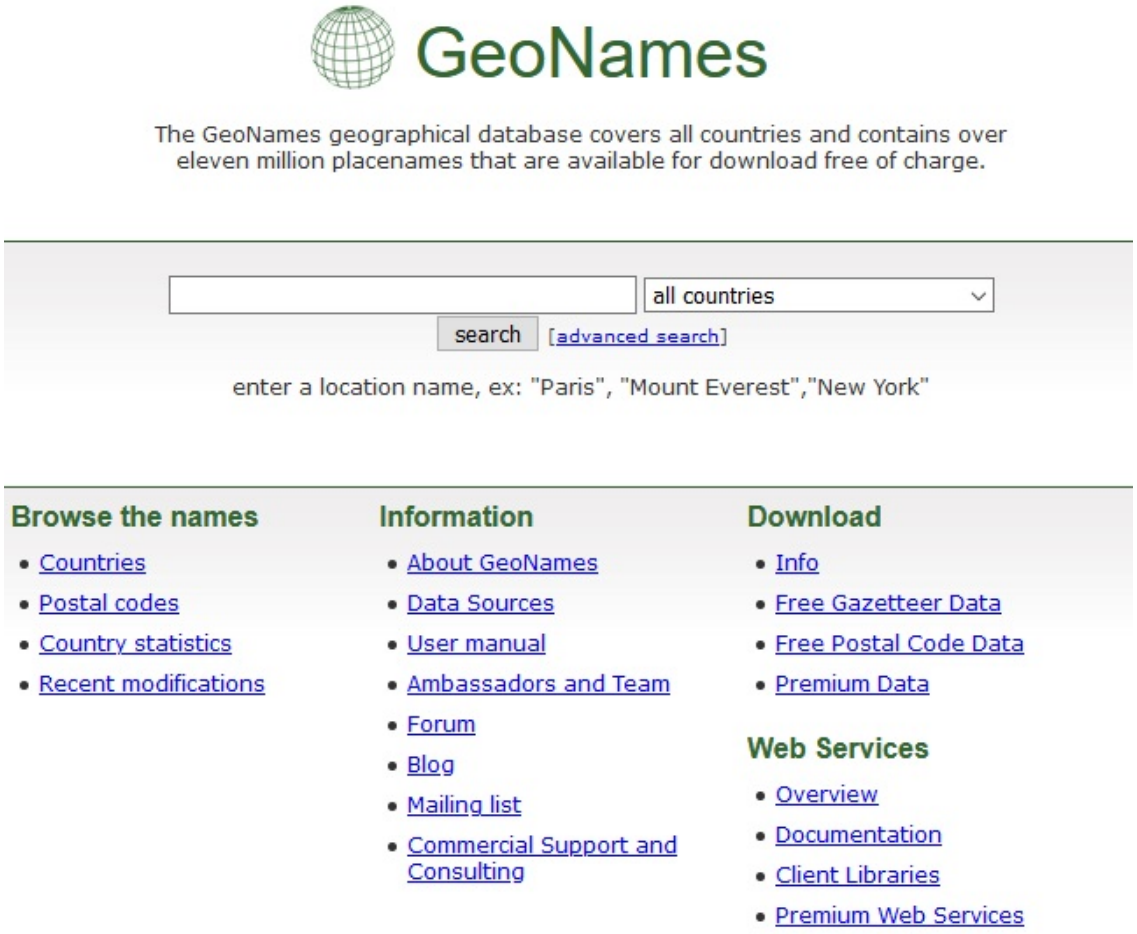


Figure: Geonames Home[19]

Data Resource Repositories

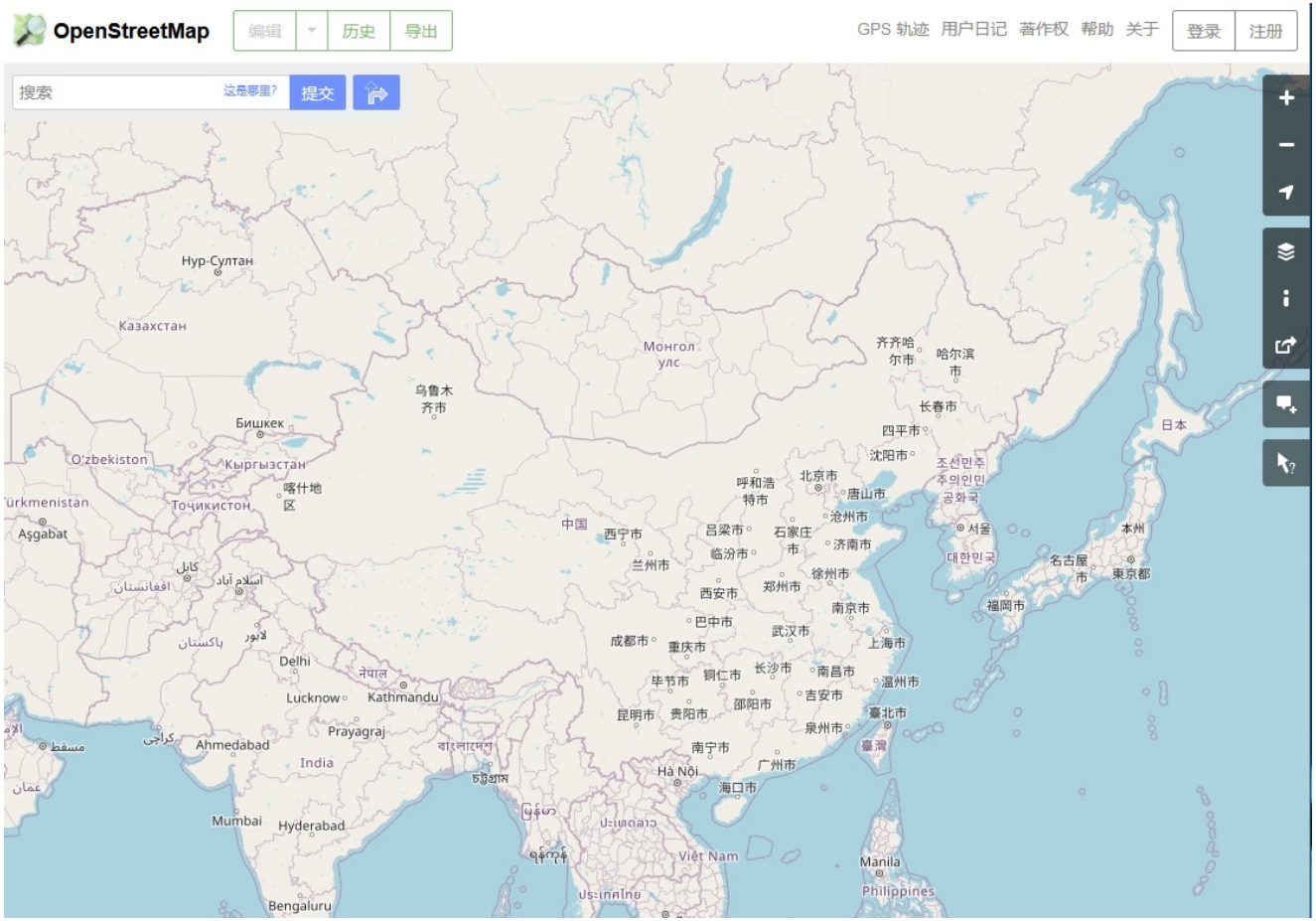


Figure: Open Street Map Home[20]

Data Resource Repositories

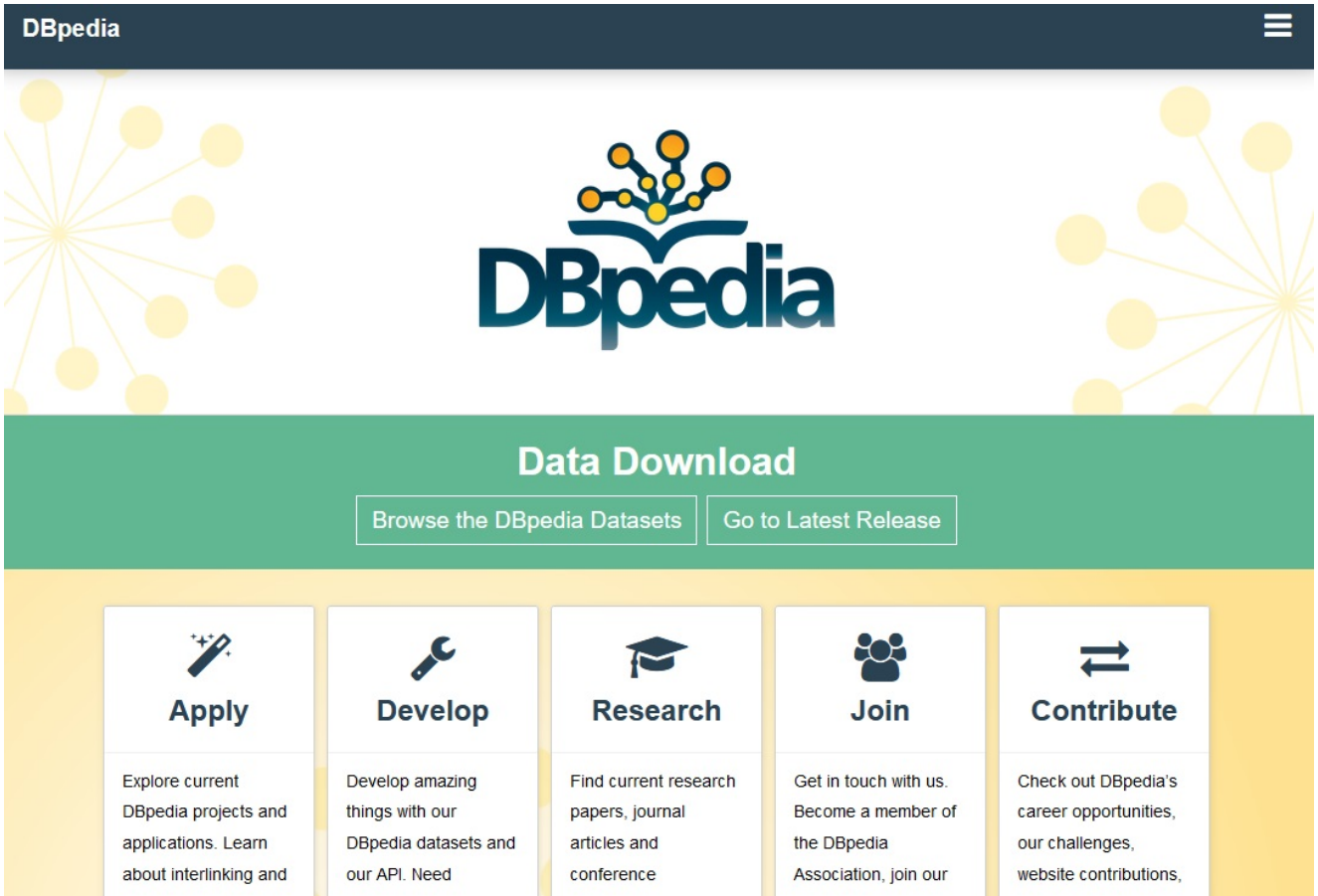


Figure: DBpedia Home[21]

[21]

Data Resource Repositories

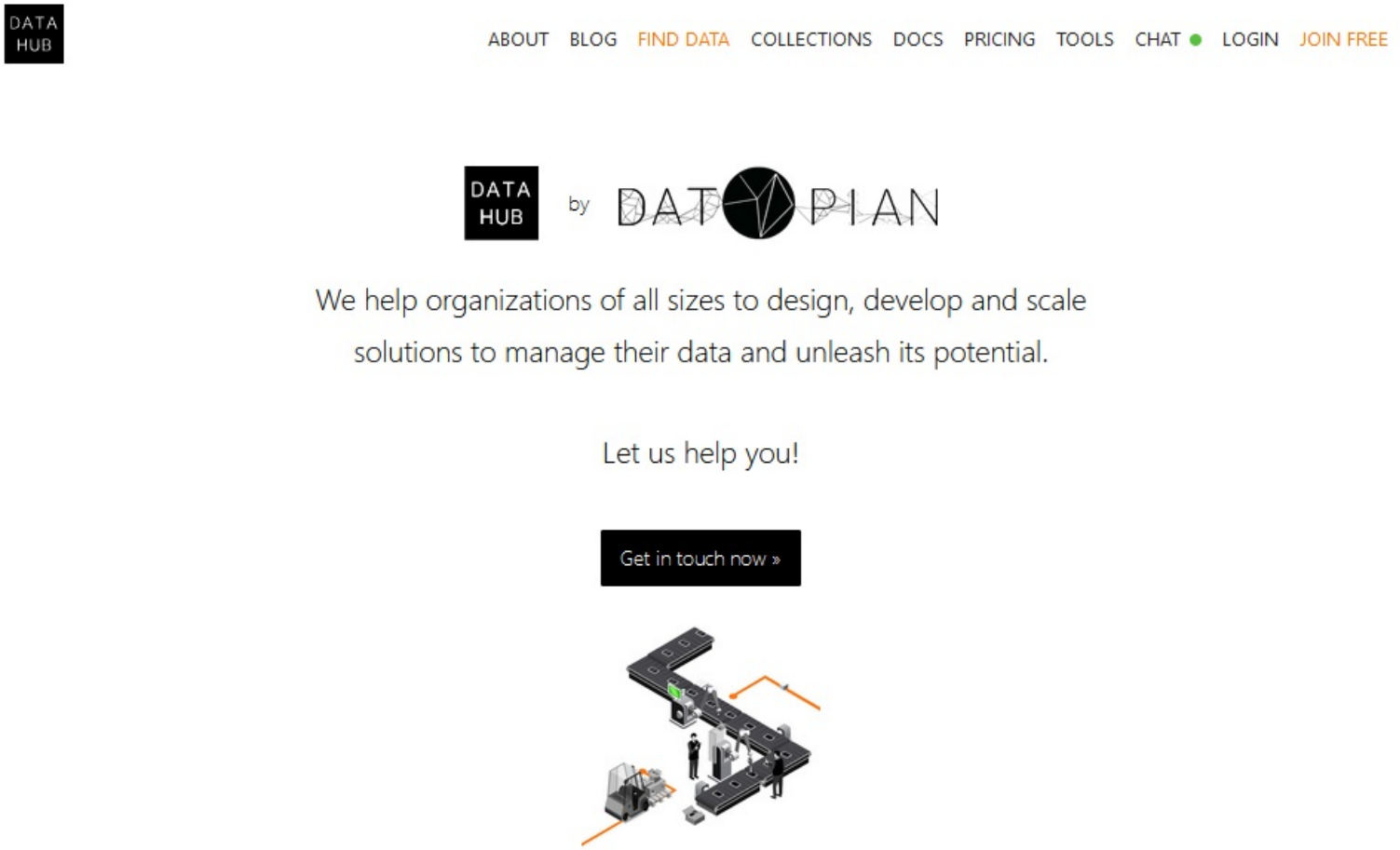


Figure: Data Hub Home[22]

Data Resource Repositories

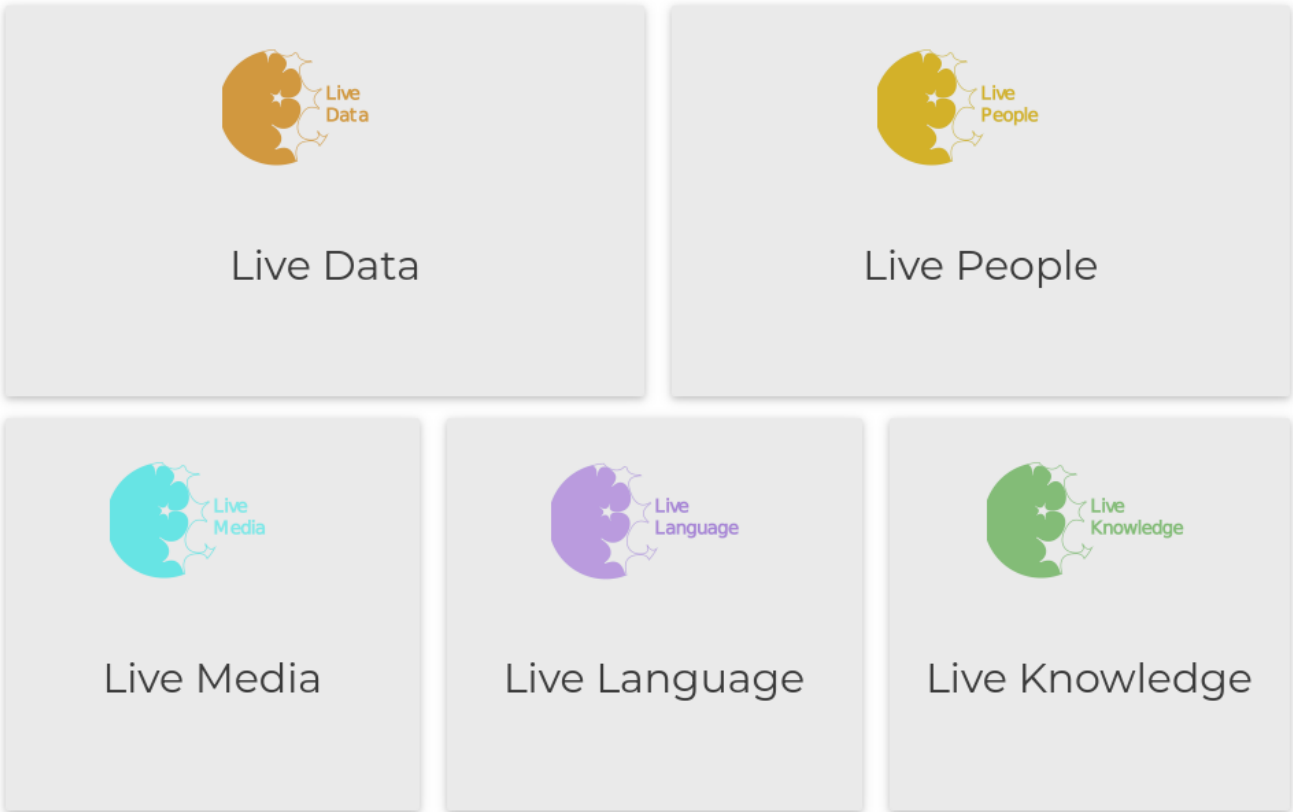


Figure: DataScientia Catalogs[23]

Part 2.3
Data Integration SoA

Data Integration (DI) - Index

Data Integration (DI) - Introduction

The language, knowledge and data resources, described above, represent the information highlighting its diversity.

Nevertheless, to exploit such a diversity, **language, knowledge and data, need to be integrated**.

The heterogeneity of the resources described above, leads to **different kinds of integration**.

- Integrate resources of the same type.
  - integrate different languages;
  - two or more, data schema, or ontologies;
  - two or more dataset.

Data Integration (DI) - Introduction

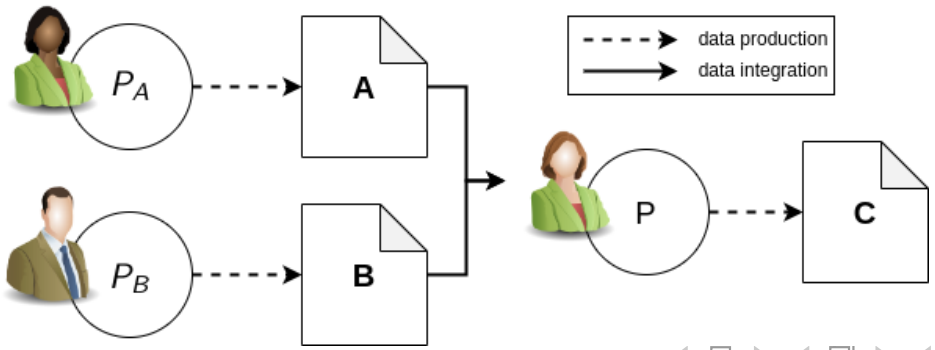
The heterogeneity of the resources described above, leads to **different kinds of integration**.

- Integrate resources of different types.
  - Integrate one dataset with a new language (different from the one used to represent its data).
  - Integrate two datasets by using a third data schema (or ontology) different from the single data schema adopted in the two datasets.
  - Integrate an ontology with a language, to produce multilingual knowledge resources.

Data adaptation and evolution

Regardless of the kind of resource integration, there two phases that always occurs when integrating resources:

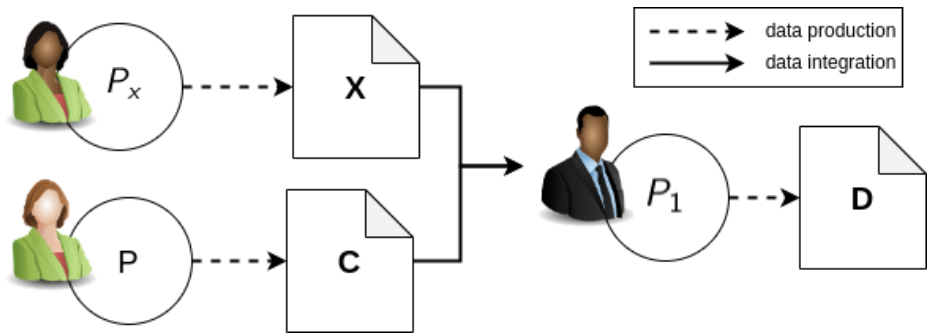
- **Data Adaptation**: this phase defines the first time that two resources, A and B, need to be integrated.
  - The resources A and B, have been created for specific purposes P_A and P_B, and they have not been modified, and/or, integrated with any other resource, to satisfy a different purpose.
  - A and B are then integrated, following a new purpose P, thus producing C as integration output.



Data adaptation and evolution

Regardless of the kind of resource integration, there two phases that always occurs when integrating resources:

- **Data Evolution:** in this phase the result of the adaptation integration of A and B, is in turn integrated to satisfy a new purpose P_1 (or an extended version of the P).
  - The adaptation integration output C, is integrated with new resources to satisfy a new purpose P_1, thus creating the result of the evolution integration D.



DI strategies

The existing DI strategies are mainly divided in two categories:

- **Virtualization strategies**: aim at providing a unique interface for two, or more, data sources, for accessing the data without extraction and transformation data.
- **Materialization strategies**: are based on ETL procedures used to extract the data to be integrated from the respective data sources.

DI Virtualization strategies - LAV

The most known set of virtualization strategies are called:
Ontology Based Data Access (OBDA)

They are all based on the modeling of an **ontology**, or a set of ontologies, interfacing the different data sources:

- **LAV - "Local As a View"** : this family of OBDA techniques assumes to query data as they are provided by the data sources.
  - single-ontology : a single ontology is queried to access all the data sources.
  - multiple-ontology : an ontology for each data source can be queried to access the data.
  - hybrid approach : as the multiple-ontology technique there are more ontologies, but in this case the queries are uniformed by a unique vocabulary used to query the data.

DI Virtualization strategies - GAV

- **GAV - "Global As a View"** : this family of OBDA techniques assumes to query data from the point of view of the application (services, or users) that needs to perform the query, by exploiting an application-specific ontology modeling.

DI Virtualization strategies - Summary

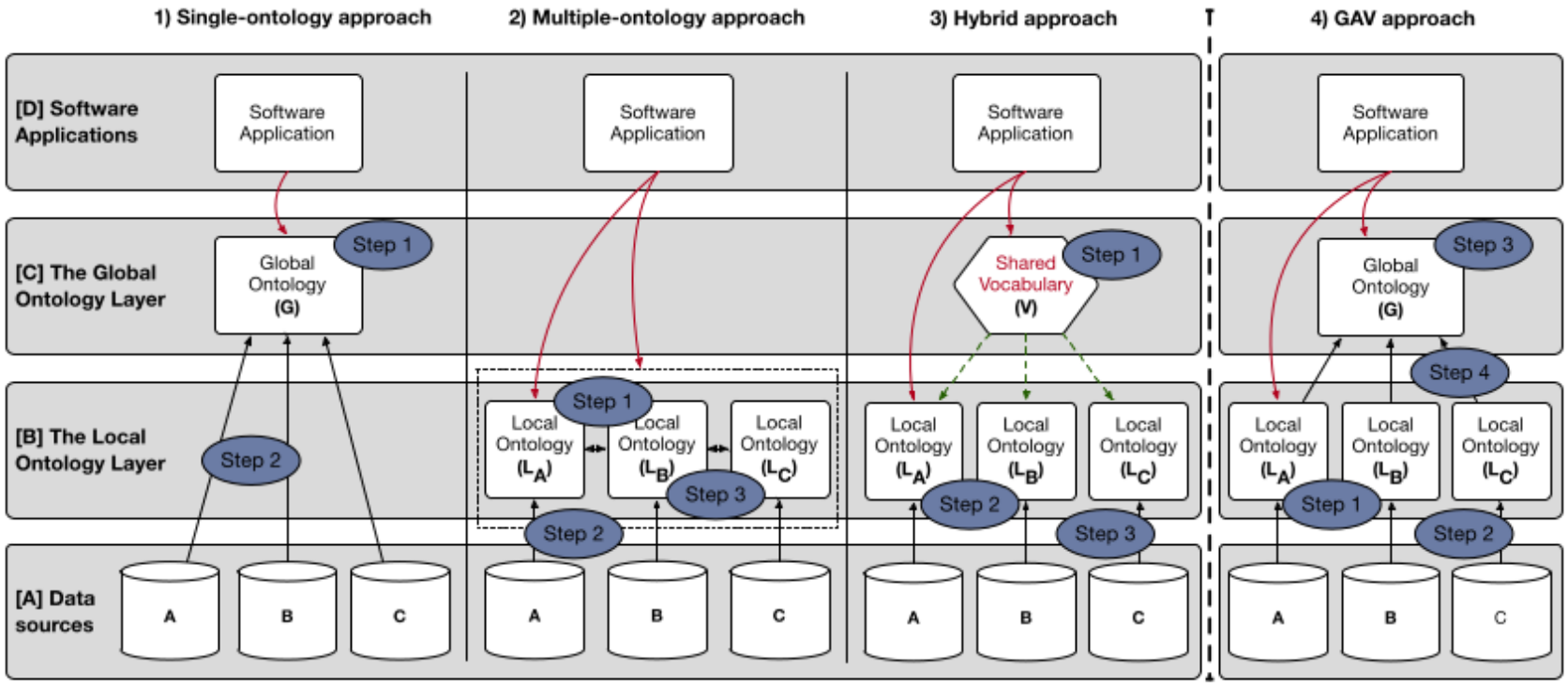


Figure 1: Three variants of OBDI from [75]: (1) single-ontology, (2) multiple-ontology, (3) hybrid, and an additional OBDI variant (4) Global-as-View (GAV).

(Explanation: Red arrows indicate access from an application to data, black arrows represent transformation/virtual access to the data; dotted green arrows represent implicit relations between involved ontologies, and numbered items show the sequence of system development. The dotted rectangle refers to the federation of local ontologies. Section 5.1 explains the additional OBDI variant (4) *Global-as-View* (GAV).)

DI Virtualization strategies - Limitations [24]

- In a GAV approach, changes in information sources, or adding a new information source, requires **revisions of a global schema and mappings** between the global schema and source schemas.
- In a LAV approach, automating query reformulation has **exponential time complexity with respect to query and source schema definitions**.

[24] Xu, Li, and David W. Embley. "Combining the best of global-as-view and local-as-view for data integration." Information systems technology and its applications, 3rd international conference ISTA'2004. Gesellschaft für Informatik eV, 2004.

DI Materialization strategies

The usage of Knowledge Graphs increased a lot within the data integration community, thanks to their suitability within different domain of interest. For this reason on of the most famous materialization strategies, is **Knowledge Graph Construction (KGC)** DI.

DI based on KGC is a process that involves different sub activities defined as follows:

- Data collection/extraction
- Schema definition/alignment
- Data cleaning & formatting
- Entity identification & mapping
- Data mapping

DI Materialization strategies - Limitations

- **Missing of** standard **methodologies** for KG generation;
  - the KGs produced are often, too application-specific, causing an increase of the KG's evolution cost.
- **Missing of frameworks** (tools and application) covering the whole KGC process.
- **Technical skills required** (data management and knowledge modeling) for the KGC process implementation;
  - the KG's final user (who is usually the domain expert) usually doesn't have such expertise.

Part 2.4
Data Architecture SoA

Data Management Architecture - Index

Data Management Architecture - Introduction

The data, before and after its elaboration (i.e., data integration), needs to be maintained by a dedicated architecture, with **the objective of serving the users, and applications, that exploit such data**.

Different kinds of data management architecture have been studied and implemented, in the past.

- Data warehouse (1960s)
- Data lake (2010)
- Data mesh (2021)

Data Warehouse

Introduced in 1960s, it is a set of centralized framework and data storage systems where:

- data is extracted from many operational databases and sources;
- data is transformed into a universal schema - represented as a multi-dimensional and time-variant tabular format;
- data is loaded into the warehouse tables;
- data is accessed through SQL-like querying operations;
- data is mainly serving data analysts for their reporting and analytical visualizations use cases.

Data Warehouse - Limitations

- Over time, they grow to thousands of ETL jobs, tables and reports that only a specialized group can understand and maintain.
- They don't let themselves to modern engineering practices such as CI/CD and incur technical debt over time and an increased cost of maintenance.

Data Lakes

Introduced in 2010, it is a set of centralized framework and data storage systems where:

- data is extracted from many operational databases and sources;
- data is **minimally transformed** to fit the storage format e.g. Parquet, Avro, etc;
- data - as close as the source syntax - is loaded to scalable object storage;
- lake storage is accessed mainly for analytical and machine learning model training use cases and used by data scientists.

Data Lakes - Limitations

- They require complex and unwieldy pipelines of batch or streaming jobs operated by a central team of hyper-specialized data engineers.
- They contain deteriorated and unmanaged datasets, untrusted and some times inaccessible, which provide little value.

Data Mesh

"*Data Mesh is a sociotechnical approach to share, access and manage analytical data in complex and large-scale environments - within or across organizations.*" [25]

A data mesh is a decentralized data architecture that **organizes data by a specific business domain**, providing more **ownership to the producers** of a given dataset.

[25] Dehghani, Zhamak. Data Mesh. Marcombo, 2022.

Data Mesh - Characteristics

- **Domain-oriented ownership**: Decentralize the ownership of sharing analytical data to business domains who are closest to the data.
- **Data as a Product**: Existing or new business domains become accountable to share their data as a product served to data users – data analysts and data scientists.
- **Self-serve Data Platform**: A new generation of self-serve data platform to empower domain-oriented teams to manage the end-to-end life cycle of their data products.
- **Federated Computational Governance**: A data governance operational model that is based on a federated decision making and accountability structure, with a team made up of domains, data platform, and subject matter experts

Part 3
The Solution - iTelos

Part 3.1
EML data representation language

Entity Modeling Language (EML)

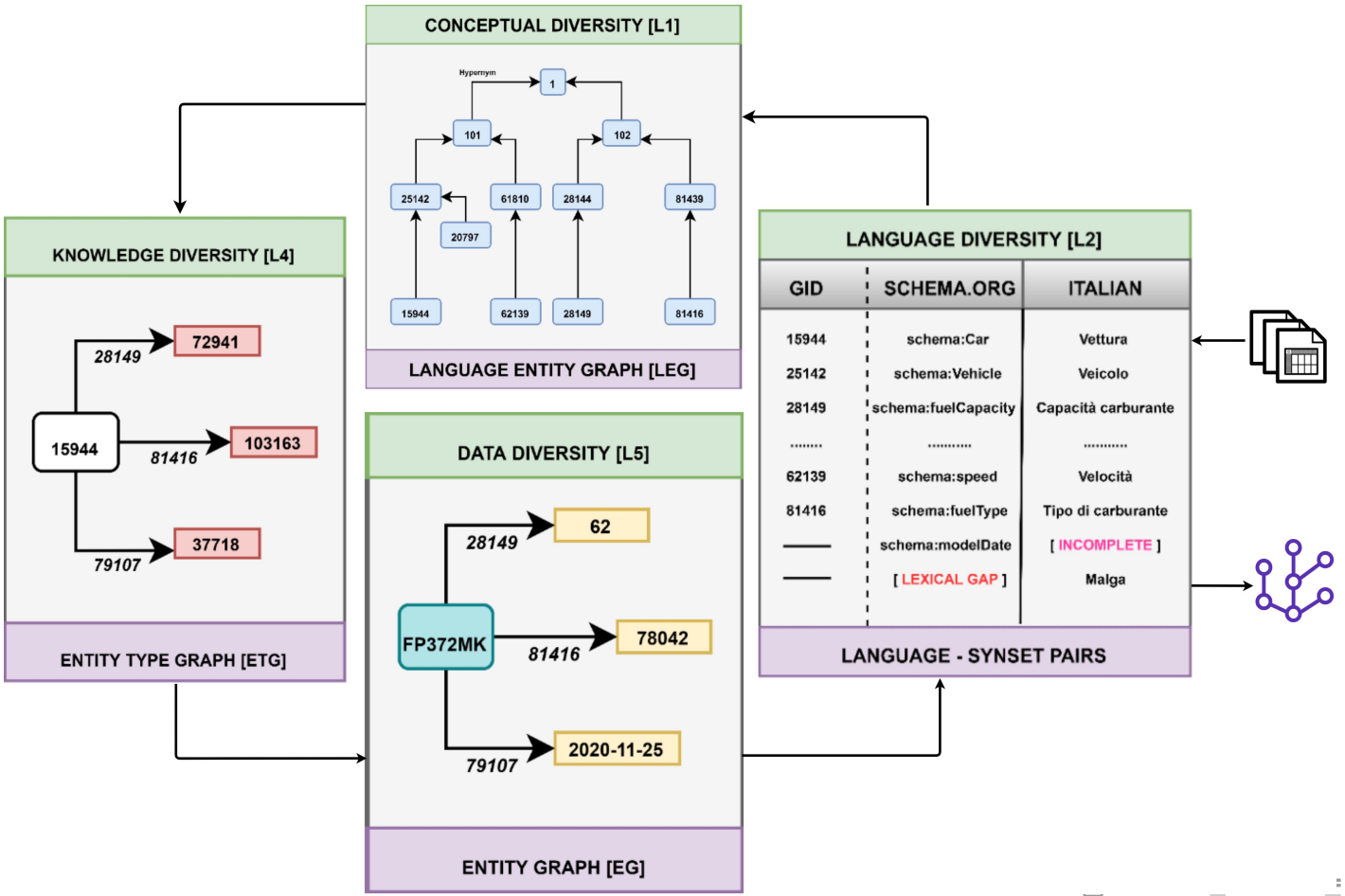
- EML is a data representation language able to **perceive and represent the data heterogeneity**.
- EML aims at **homogenising the heterogeneity of data by providing unique representations** for all four levels of heterogeneity.
  - EML-Sc homogeneity in Source heterogeneity
  - EML-F homogeneity in Format heterogeneity
  - EML-S homogeneity in Structure heterogeneity
  - EML-M homogeneity in heterogeneity

EML - Stratified Approach

- EML is designed over a **stratified information approach** in which the information is always composed by three layers of resources.
- Such layers are those already discussed as types of reusable resources
  - (L) Language (and concepts) resources
  - (K) Knowledge resources
  - (D) Data values resources

EML - Stratified Approach

- The stratified approach defines **how the information is a represented as a composition of different resources of different layers**.



Entity Modeling Language (EML)

- Therefore, EML handles the data heterogeneity by considering two orthogonal dimension: **the type of heterogeneity, and the information layers**.
- This means that EML defines homogeneity for:
  - EML-Sc(L), EML-Sc(K), EML-Sc(D)
  - EML-F(L), EML-F(K), EML-F(D)
  - EML-S(L), EML-S(K), EML-S(D)
  - EML-M(L), EML-M(K), EML-M(D)
- Let's discover more concretely how EML define all the above data homogeneity components.

EML-Sc

- As already discussed "*Source heterogeneity refers to the divers possible sources of information from which information resources can be collected*".
  - The heterogeneity is caused by the different interpretations of the information that each source has, over the same target reality.
- EML-Sc aims at homogenizing such heterogeneity, by **selecting reliable and standardized sources**, thus **limiting noisy interpretation of the target reality** respect to a specific purpose.
- Concretely EML-Sc is the **set of information sources** from which the project resources can be collected.
  - Websites
  - Data catalogs
  - Databases and Knowledge bases
- The set of sources defined by the EML-Sc is applied over Language, Knowledge and Data resources, thus defining in specific the sub-sets: **EML-Sc(L), EML-Sc(K), EML-Sc(D)**.

EML-F

- *"Format heterogeneity refers to the divers possible data formats that can be employed to differently encode information"*.
- EML-F aims at homogenizing such heterogeneity for all the three types of information (Language, Knowledge and Data), by **defining which standardized and well-known formats have to be applied to represent the information to be reused**.
  - **EML-F(L)**: Excel, XML
  - **EML-F(K)**: RDF-OWL [26] [27]
  - **EML-F(D)**: JSON, Excel, CSV, RDF

[26] RDF documentation
[27] OWL documentation

EML-S

- *"Structure heterogeneity is conventionally understood as the existence of variance in the representation and description of the same target reality"*.
- EML-S aims at homogenizing such heterogeneity by **defining the structures of the information elements** for each types of information (Language, Knowledge and Data).
  - **EML-S(L)**: defines the *concept* structure [word, sense, synset, concept] - UKC structure
  - **EML-S(K)**: defines the *EType* structure [data and object properties, annotations]
  - **EML-S(D)**: defines the values *data types* [string, int, long, bool, ...]

EML-M

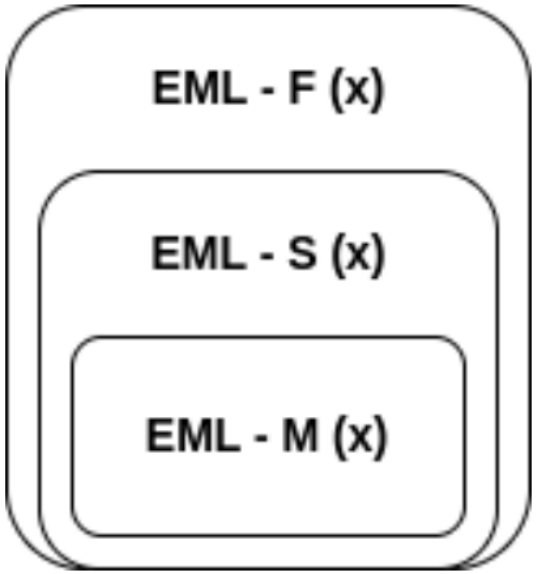
- *"Meaning Heterogeneity, is defined over the values of the information properties which can be used to identify a real world entity"*.
- EML-M aims at homogenizing such heterogeneity by **defining which values have to be adopted to identify real world entities, and how to shape such identifiers**.
- For each types of information, EML-M defines the required identifiers.
  - **EML-M(L)**: concept ID
  - **EML-M(K)**: EType ID
  - **EML-M(D)**: SURI/SURL

EML outcome

- The homogeneity introduced by EML over the resources to be reused, **reduces the cost to exploit thus resource in future** (reusability), due to:
  - the reduction of the heterogeneity,
  - the adoption of standards formats and information structures, and,
  - the adoption of identifiers.

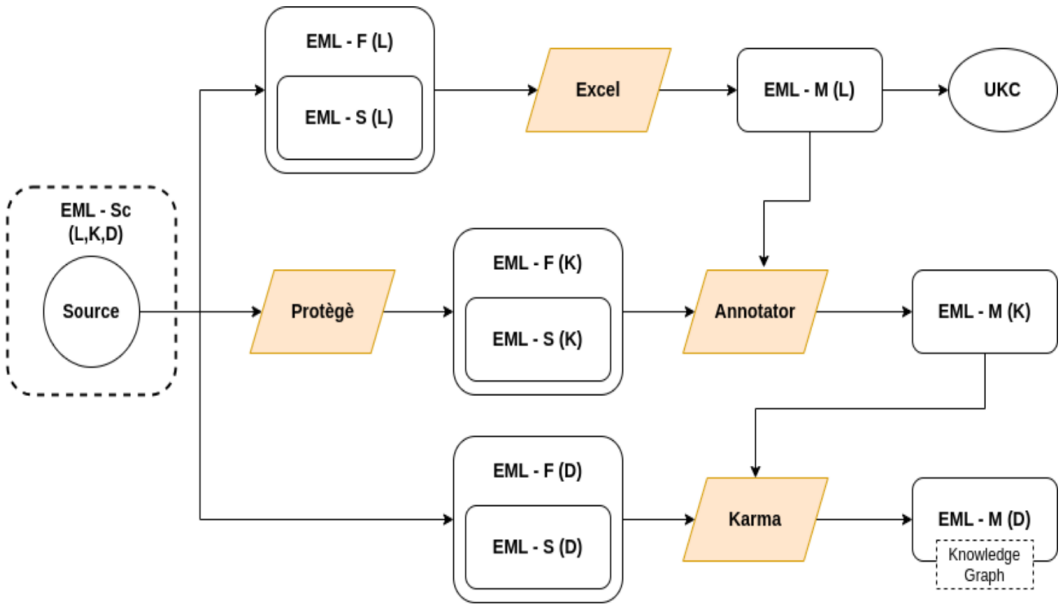
Progressive Ordered Encapsulation (POE)

- It is important to notice how the homogeneity introduced by EML is **progressively encapsulated starting from the information sources until the generation of EML-compliant reusable resources**.
  - This means that EML-M includes the homogeneity defined in EML-S, as well as, EML-S includes the homogeneity of EML-F, and so on.



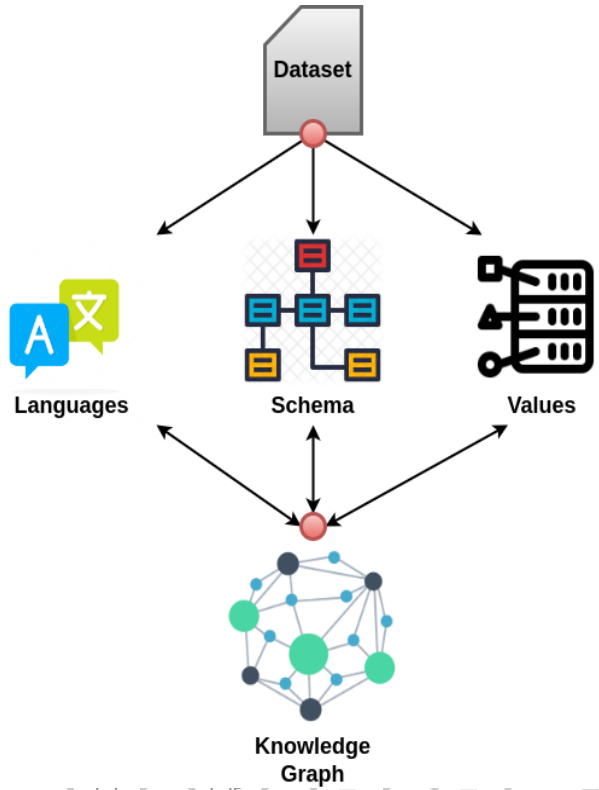
EML generation process

- The generation of EML-compliant resources is a responsibility of the **data reuse processes** that will be described in the next section.
- Nevertheless we can report here which is the EML generation process that will be considered in the top level processes, together with the tools allowing for the EML generation.



EML standard

- The different layers extracted (defined explicitly) from "conventional" datasets are **composed together** into a **Knowledge Graph** (KG).
- Why a Knowledge Graph ?
  - adaptability (different contexts)
  - scalability
  - its structure allows:
    - the information **layers composition**, and,
    - the **KG decomposition** into single layers resources.



Part 3.2
iTelos data reuse processes

iTelos Reuse Processes - Data Flow

Low quality and low reusability of data make the reuse of data time-consuming and costly. Thus enhancing the cost of the communication between *Producer* and *Consumer*.

To reduce such costs, the idea is to introduce a new agent as mediator between producer and consumer, having the objective of handling data to make them reusable and interoperable:

- **The Data Intermediary**

Data Intermediary

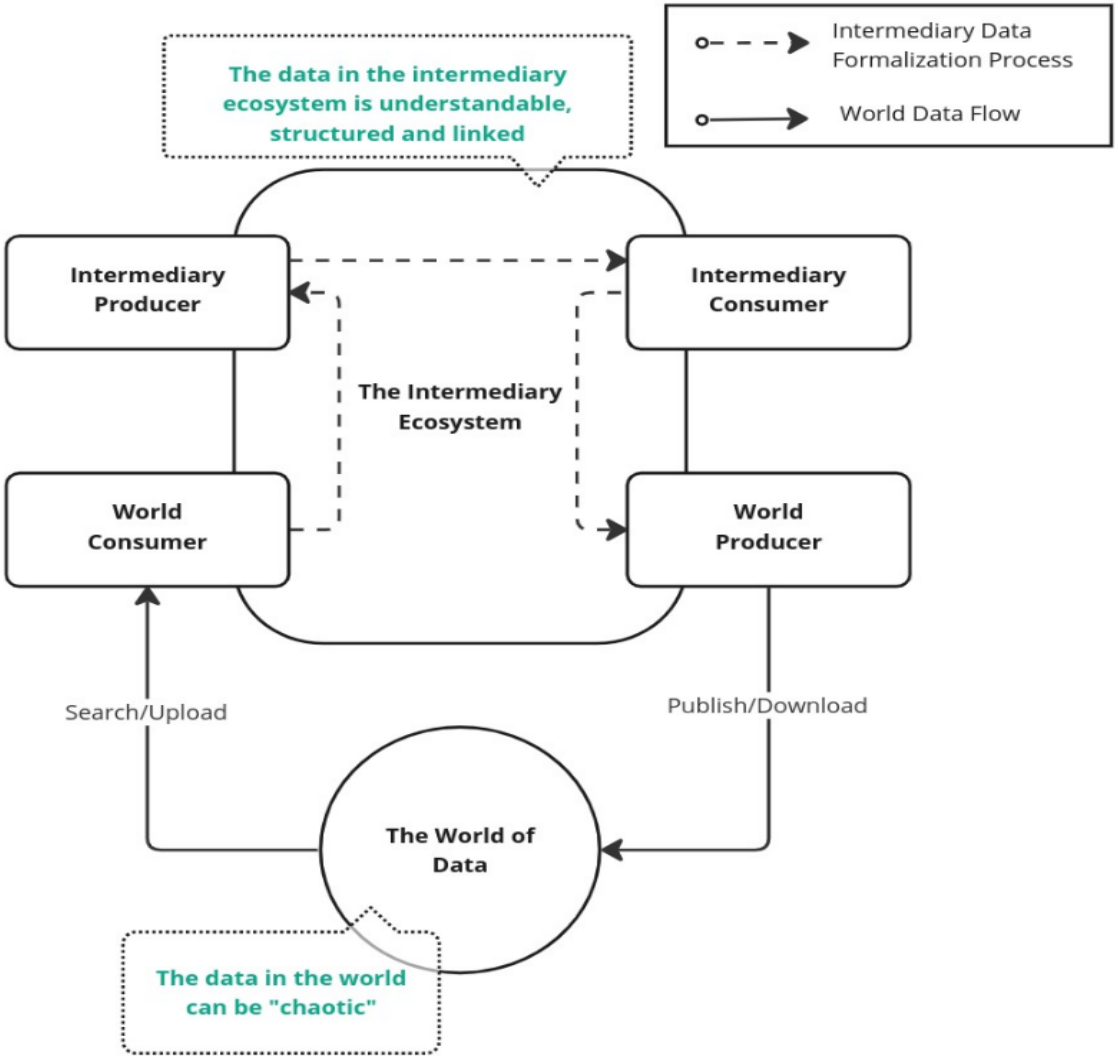
- **Definition**: A data intermediary is a *human agent* who supervises a structured *process* for the production of *language* specific resources, implemented into a dedicated *environment*.
  - Thus, DI := [language + process + environment].
- **Objective**: A data intermediary aims at:
  - collecting data from the (disordered) heterogeneous world of data;
  - cleaning and formatting following well-known standards;
  - increasing the reusability of such data;
  - generating purpose-specific reusable data to support specific application and services;
  - sharing high quality and reusable data to enhance homogeneity into the heterogeneous world of data.

Data Intermediary

To achieve its objectives, the data intermediary **agent** is internally divided in:

- **Intermediary Producer**: it collects and transforms low quality data into high quality and reusable resources.
- **Intermediary Consumer**: based on a specific purpose, it composes the high quality resources, produced by the intermediary producer, to support purpose-specific application and data services.

Data Intermediary Action



Data Intermediary

- Concretely the Data Intermediary is composed by:
  - The EML language,
  - The iTelos data reuse processes.
  - The Distributed Stratified Data Mesh.
- While the EML have been already described above, the below sections will be focused on the remaining two components.

iTelos Dat Reuse Processes

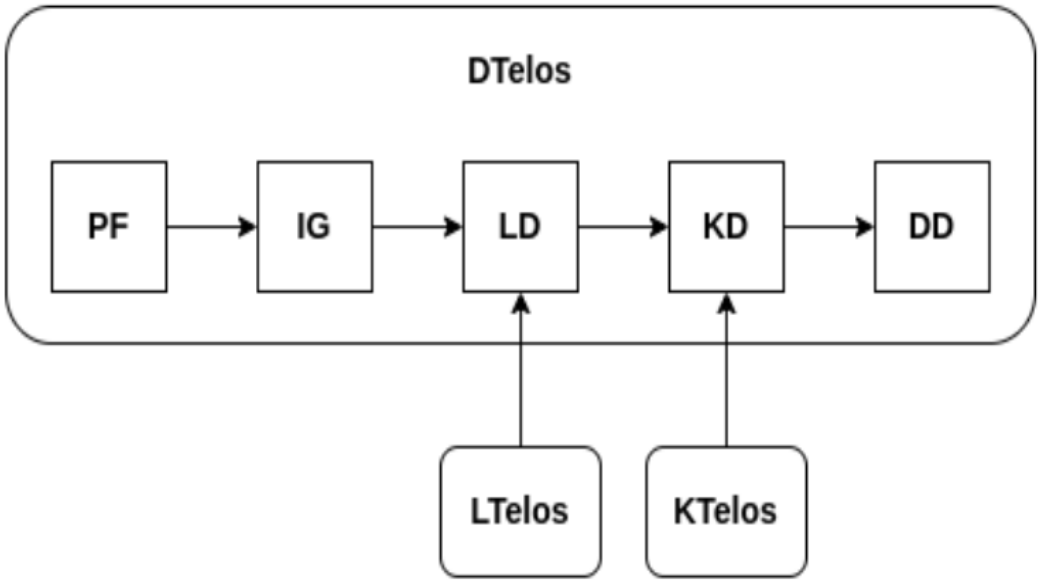
- **iTelos** is a **structured, phase-based methodology, implemented by three parallel process that share the same methodology structure**.
  - The three processes are dedicated to the three types of information respectively.
    - **LTelos**: generation of EML-M(L) resources
    - **KTelos**: generation of EML-M(K) resources
    - **DTelos**: generation of EML-M(D) resources

iTelos Dat Reuse Processes

- Due to the POE feature of EML, the progressive encapsulation is transferred over the iTelos process too.
  - For this reason **the DTelos process involve the (parallel) execution of KTelos, as well as, KTelos involve the execution of LTelos.**
- **Note**: For lack of time, the KGE course is focused on DTelos, assuming the other two processes executed in parallel.

iTelos Data Reuse Process - DTelos

- Below the DTelos process structure, **based on the iTelos methodology**, where the LTelos and KTelos processes are taken in input (see next slide for the details over each single process phase).



iTelos Methodology

iTelos [28] is a phase-based methodology (implemented by LTelos, KTelos and DTelos) that,

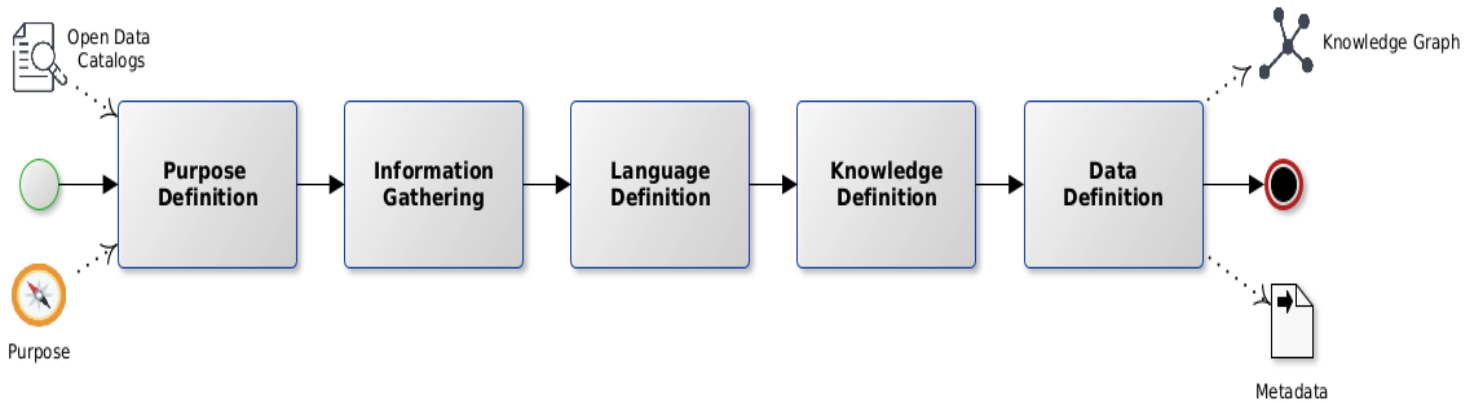
- takes in **input** a set of data resources, having an arbitrary level of quality.
- The methodology structure is the same for:
  - **Intermediary Data Producer (IDP)**: collecting and transforming existing resources into reusable (EML-compliant) resources.
  - **Intermediary Data Consumer (IDC)**: composing already produced EML-compliant resources.
- And produce as **output** high quality data, shaped as:
  - (IDP) different knowledge graphs (e.i., one for each input dataset);
  - (IDC) a single knowledge graph created by composition of existing KGs and/or lower quality resources.

[28] From greek "**telos**" which means purpose

iTelos Methodology

iTleos is structured in 5 well-defined phases, summarized here below [29]

- Purpose Definition (PF)
- Information Gathering (IG)
- Language Definition (LD)
- Knowledge Definition (KD)
- Data Definition (DD)



[29] the phases will be detailed in Part 4 - iTelos methodology

Purpose Definition

- Both data producer and consumer consider their own objective when building KGs.
- Such an objective implicitly includes the user "point of view", the representation that the user uses to model (a portion of) the world, where the desired information lives.
- We define the user objective, **The Purpose** which will lead the entire KGE process.

Purpose Definition

- **Input**: a natural language sentence representing the user's Purpose (plus optionally a list of already identified data sources to be exploited).
- **Output**: a set of documents and models in which the Purpose's requirements are extracted and formalized.
- **Objective**: to formalize the functional requirements implicitly included in the input user's purpose.

Information Gathering

- **Input**: a set of data sources identified previously, plus the formalized user's purpose.
- **Output**: a set of resources collected from the input data sources, suitable to satisfy the purpose.
- **Objective**: the second phase of iTelos aims at collecting the resource, to be processed, with the objective to build the final KG(s)

Information Gathering

- The gathering of information includes the collection of resources **for all the stratification layer**: data, knowledge and language.
- Notice how, depending by the agent that executes the process, the resources collected have different levels of quality:
  - intermediary producer: the resources are collected from the "disordered world", thus the quality level is, in average, lower.
  - intermediary consumer: the resources are collected from the "ordered world", thus the quality level is, in average, higher.

Language Definition

- In this phase, iTelos aims at defining the "**language of the KG(s)**".
  - concepts and terms used to define the information to be exploited
- Notice that the information in the KG(s) could be defined by using not only **natural languages** but also **domain languages**.
  - standard concepts and terms defined for a specific domain (e.i, healthcare standards, unit of measure codes).
- The language definition phase is supported by the UKC project [30]

[30] **TODO UKC ref**

Language Definition

- **Input**: the resources collected previously, plus the formalized user's purpose.
- **Output**: a set of language resources defining the concepts and terms to be adopted to define the KG(s) information.
- **Objective**: the third phase of iTelos aims at fixing the right concepts and terms for the KG(s)'s information, thus reducing the semantic heterogeneity of the final outcome.

Knowledge Definition

- Once the information is clearly defined by fixed concepts and terms, it needs to be **structured**.
- The modeling of the knowledge layer of the KG(s) **unifies the representation** of the information handled by the KG(s)
- iTelos models the KG(s)'s structure by exploiting a precise knowledge modeling methodology [31] (detailed in the next chapter) based on the ontology and teleology theory.

[31] **TODO kTelos ref**

Knowledge Definition

- **Input**: the resources previously collected (knowledge and data) and produced (language), plus the formalized user's purpose.
- **Output**: one, or a set of (one for each KGs to be produced) knowledge resources.
- **Objective**: the knowledge resources produced in this phase aims at:
  - unifying the representation of the information;
  - improving the **interoperability** and **reusability** of the final KG(s), by building knowledge resources reusing as mush as possible well-known standard domain ontologies and data schema.

Data Definition

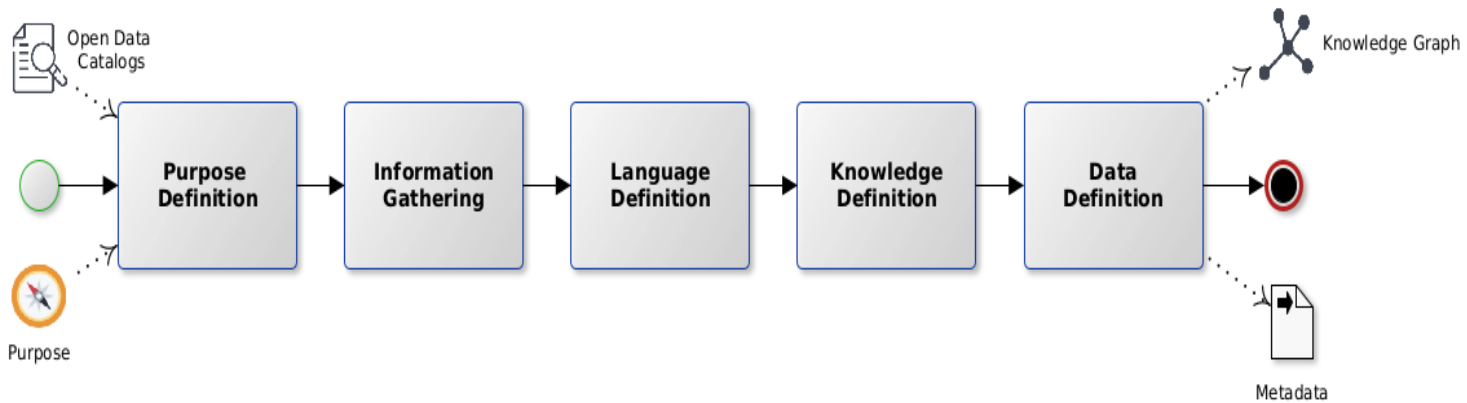
- **Input**: the resources previously collected and produced (knowledge, language and data), plus the formalized user's purpose.
- **Output**: the final KG(s).
- **Objective**: the last phase of the methodology aims at merging the knowledge resources previously defined, with the cleaned and formatted data to be considered by the KG(s), thus producing the final concrete outcome.

Data Definition

- The last phase of the methodology is dedicated to the data layer of the final KG(s).
- It is supported by a specific data mapping tool.
- How it will be better detailed in the next chapter, in this phase there two activities plying a crucial role:
  - Entity recognition: to find the real world entity within the dataset collected.
  - Entity matching: to disambiguate different representations of the same real world entity.

General methodology characteristics

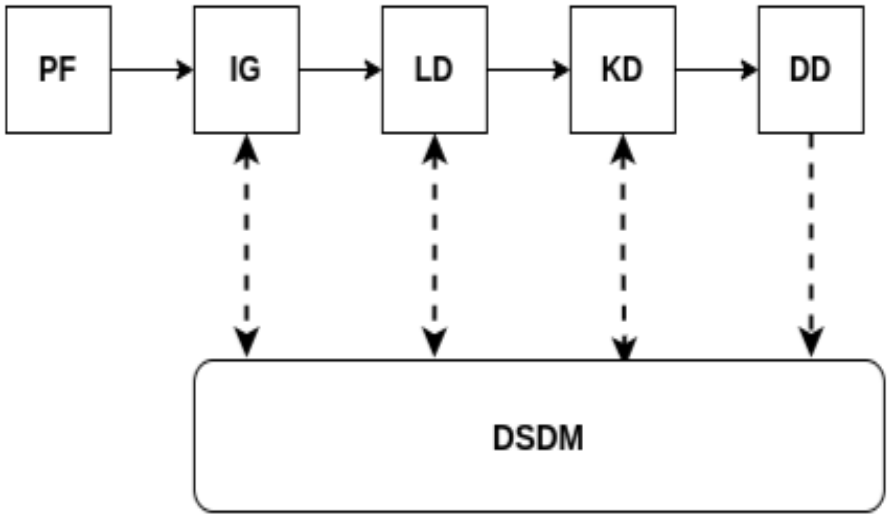
- The iTelos phases are structured following the **stratified approach**.
- iTelos builds the KG adopting a **middle-out approach** between knowledge and data, so that it is
  - not too much focused on the knowledge layer (top-down approach);
  - neither too much focused on the data layer (bottom-up approach).
- For each iTelos phase, it exists an activity of **metadata definition**.
  - such activities define a parallel process which aims at producing metadata for the different resources composing the KG(s), for data **distribution purpose**.



Part 3.3
Distributed Stratified Data Mesh (DSDM)

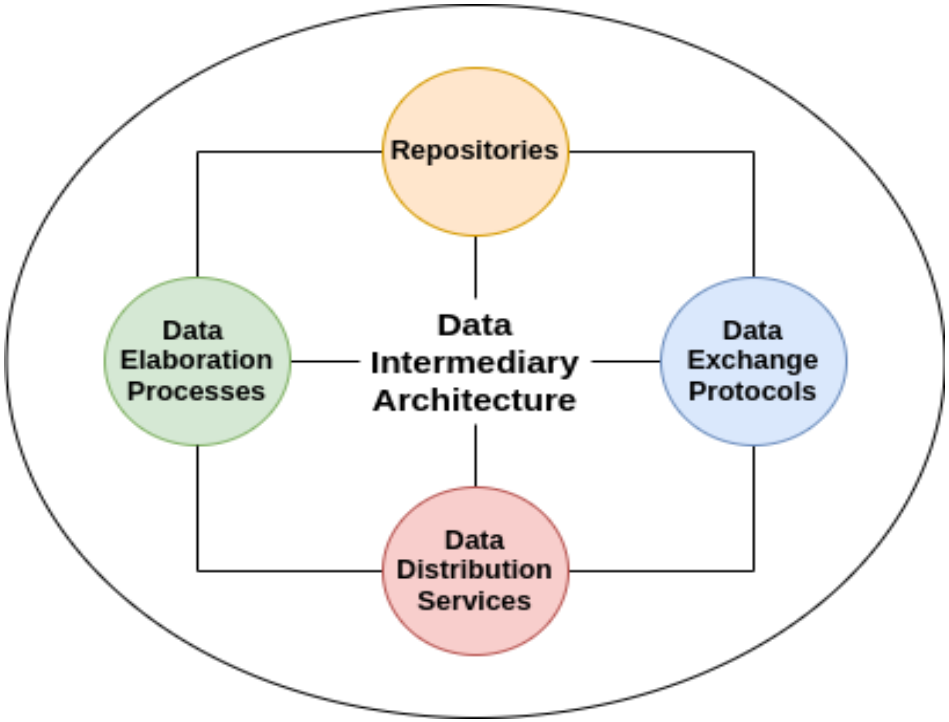
Data intermediary architecture

- To implements the iTelos methodology, thus supporting LTelos, KTelos and DTelos, the data intermediary needs a dedicated data architecture, called **The Distributed Stratified Data Mesh (DSDM)**.



Data intermediary architecture

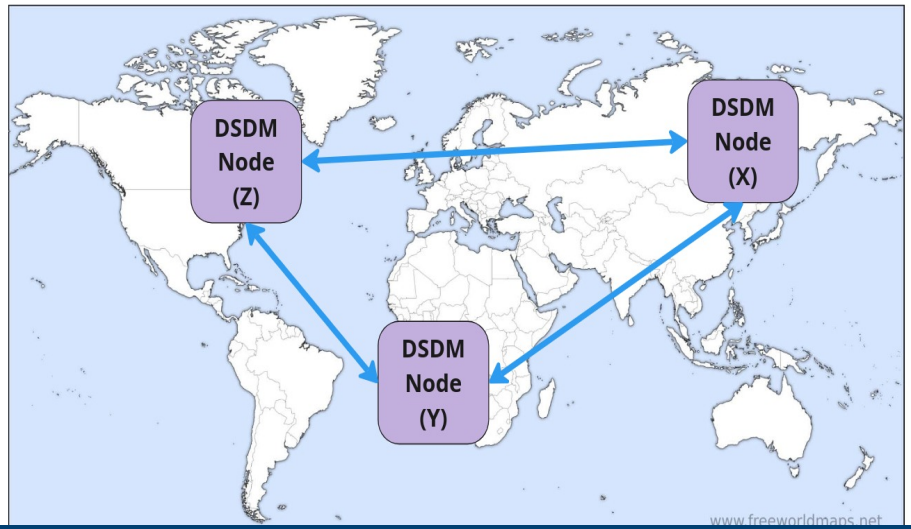
- The DSDM is defined by **different components communicating each other**.



Distributed Stratified Data Mesh (DSDM)

More in details the intermediary data architecture is a Distributed Stratified Data Mesh (DSDM);

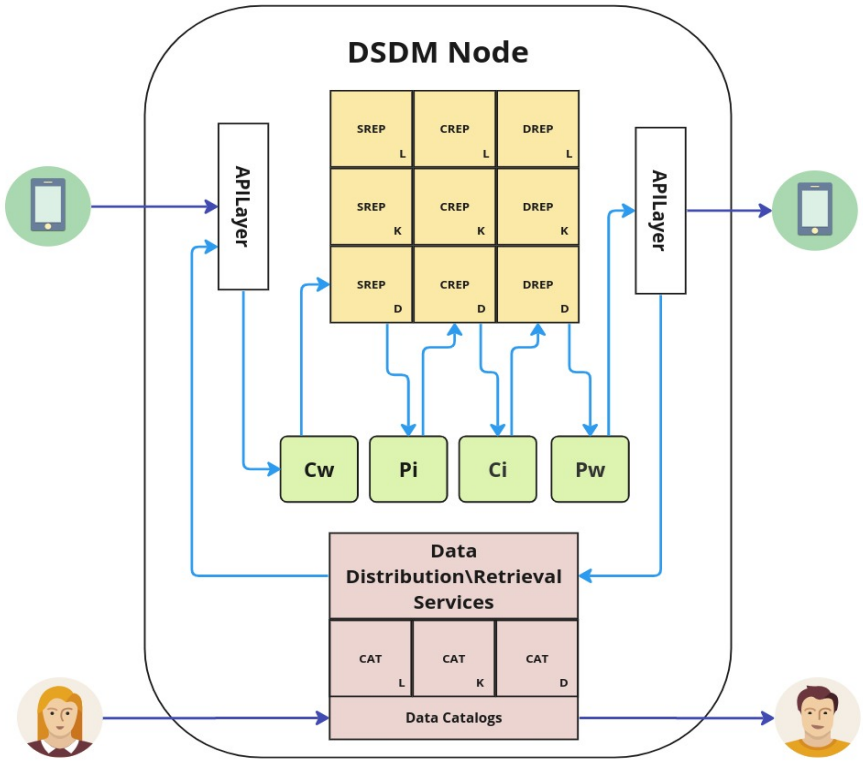
- it includes all the data mesh features;
- additionally, it is composed by different **nodes**, where each node;
  - is defined for a **specific domain** of interest, or purpose (i.e., geographically);
  - **autonomously manages resources** about its domain/purpose (local domain experts handling data);
  - has a **local implementation** of the intermediary data architecture;
  - For each node, **it handles stratified resources** (Language, Knowledge and Data)



Distributed Data Mesh - The Node

Each node of the DSDM
is composed by the following components
(Detailed in the following slides):

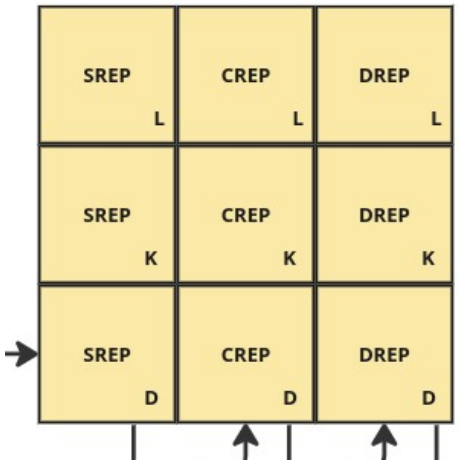
- Data repositories
- Data elaboration processes
- Data exchange protocols
- Data distribution services



Data Repositories

Each DSDM node includes three different repositories, which are distinguished on the basis of what data they contain.

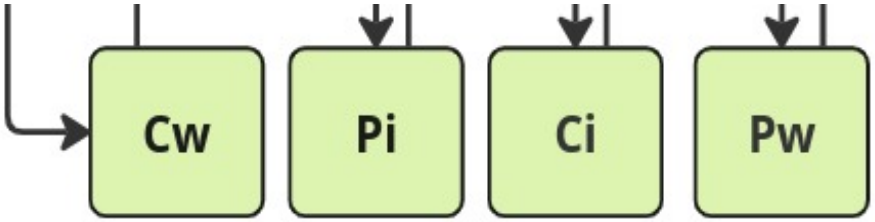
- **SREP**: the Source REPository stores the data collected form the "disordered" world, which need to be processed to make it compliant with the data intermediary data requirements about quality and reusability.
- **CREP**: the Core REPository stores the data that has been processed by iTelos, thus being compliant with the intermediary data requirements.



- **DREP**: the Distribution REPository stores the data which can be accessed by the data distribution services. In other words, the data which can be shared out of the DSDM node.

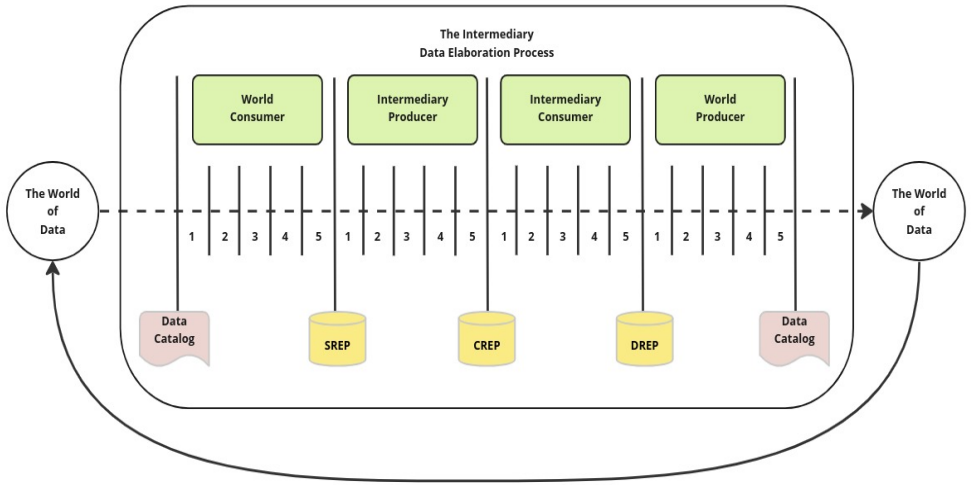
Data Elaboration Processes

- The data elaboration processes are responsible for the **collection and "transformation"** of the data, that before and after each process are stores in one of the above described repositories.



Data Elaboration Processes

- The data elaboration processes are **four different instances of the DTelos process** [32].
- Depending on the objective to be achieved, DTelos can be adopted, by exploiting the features offered by its different phases.
  - Some phases are more (or less) exploited then others depending by the elaboration process which need to be executed.



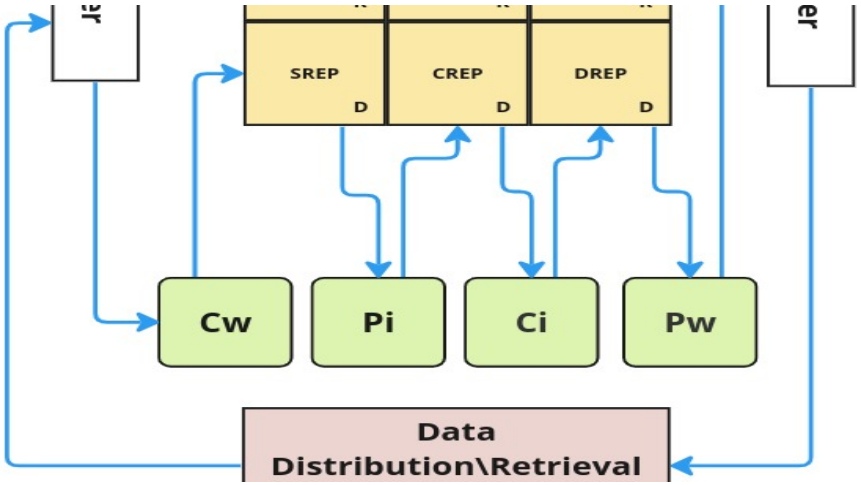
1. **Purpose Formalization**
2. **Information Gathering**
3. **Language Definition**
4. **Knowledge Definition**
5. **Data Definition**

miro

[32] different problems, one methodology

Data Exchange Protocols

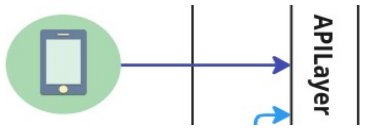
- The data exchange protocols are distinguished in two types:
  - **Internal protocols**: they define the exchange of data, **within the DSDM node**, between repositories, processes and data distribution services.



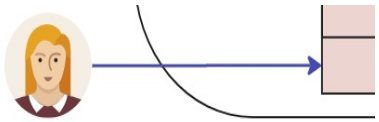
Data Exchange Protocols

- **External protocols**: they define the exchange of data, **across different DSDM nodes**, by considering

- automatic data exchange (device to device)

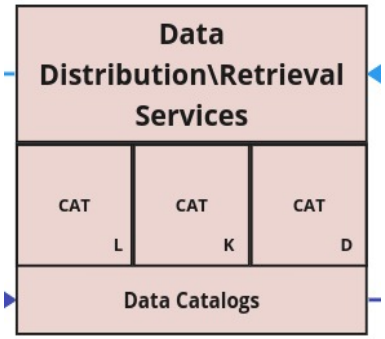


- human-driven data exchange (human to device)



Data Distribution Services

- The data distribution services aims at **sharing the resources** produced, and handled, by the DSDM node.
- Such services plays a crucial role in the **reusability** of the data accessible in the whole DSDM.
  - The exploitability of the data increases.
  - The effort in building new (EML-compliant) quality data, decreases.



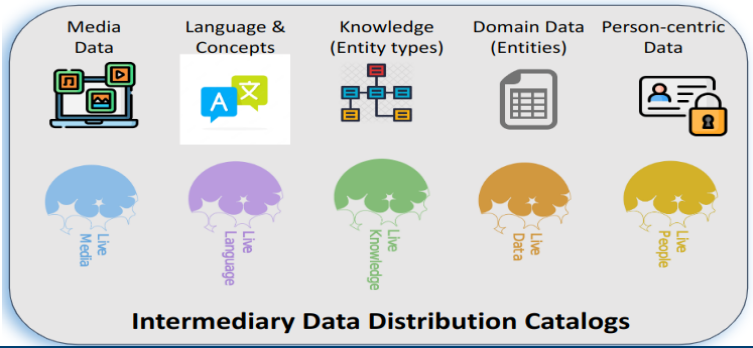
Note: more details about the intermediary data distribution are provided in the following slides.

Intermediary Data Distribution

- Within the data intermediary architecture, as already mentioned above, a crucial role is played by the data distribution.
- The importance of such component in the architecture, leads to the definition of a **dedicated architecture** for the data distribution.
  - **The Distributed Metadata Mesh**
- The metadata mesh is directly **mapped over the DSDM nodes**, but it handles **metadata** describing the data that has to be shared.
  - This allows the whole intermediary architecture to:
    - **provide information** about the data to be shared, through their metadata;
    - **protect** the data, and distribute it only once it needs to be exploited.

Distributed Metadata Mesh - Nodes

- The single node of the metadata mesh, enables the exploitation of the data distribution (and retrieval) services.
- To this end, a single node is composed by 4 resources catalogs, the we can divide in two categories:
  - **data catalogs**: these catalogs shares metadata about domain data and person-centric data. Their names are **LiveData** and **LivePeople**, respectively. (Media data will be considered in future)
  - **interoperability catalogs**: these catalogs shares metadata about language and knowledge resources, which can be used to represent the data. The resources considered by these catalog are provided to enhance the **interoperability** of the data. The catalogs have been called, **LiveLanguage** and **LiveKnowledge** respectively.



Distributed Metadata Mesh - Catalogs Links

- The KG(s) produced by DTelos, within each DSDM nodes, are **stratified**, thus composed by language, knowledge and data resources.
  - The link among the different resources, composing a KG, is maintained also at metadata level.

Example: a data resource, shared in LiveData, has a specific metadata linking to the knowledge resources, in LiveKnowledge, defining the initial data schema. Using the same approach the knowldge resources is linked to one (or more) language resources in LiveLanguage.

Linked data catalog example

Distributed Metadata Mesh - Navigation entry point

- The distributed data mesh can be accessed by the users who wants to navigate it, through the catalogs webportals.
- The entry point for such catalog navigation, is the **Main LiveData** catalog.
  - This "top-level" catalog, unlike the others catalogs, collects metadata about the the local LiveData catalogs, providing the direct access to them.

Distributed Metadata Mesh - Services

- The metadata mesh provides a set of services which can exploited through the catalogs. Here below the list of the available services.
  - **Catalog deployment**: offered by the Main LiveData catalog, it allows new organization and/or users, to easily deploy a new domain specific data LiveData catalog (service that can be adopted also for the other types of catalogs).
  - **Data Upload**: offered by each catalog, it allows the user to upload new data, in the relative DDM node (notice that such data is still not published).
  - **Data Publication**: offered by each catalog, it allows the user to publish a set of metadata for a new resource (already available in the DDM node repositories) to be shared.

Distributed Metadata Mesh - Services

- **Data Search**: offered by each catalog, it allows the user to find resources by searching for its metadata values.
- **Data download**: offered by each catalog, in a different way depending by the data access policy defined by the DDM node, this service allows the user to download the data required.
- **Data composition on demand** (under development): this services will allow the users to select and compose the resources (language, knowledge and data) they need to build a new KG.

Part 4
The iTelos Methodology

Intermediary Producer

- objective - data production (from informal stream data to formal data)
- input & output
- overall producer methodology

Intermediary Consumer

- Objective - data composition purpose specific
- input & output
- overall consumer methodology

Phase 1 - Purpose Definition

Purpose Definition - Producer

- functional req.
- non-functional req. Data Production
- ER model

Purpose Definition - Consumer

- functional req.
- non-functional req. Data Composition
- ER model

Phase 2 - Information Gathering

Information Gathering - Producer

- sources:
  - informal resources
  - semi-formal resources
- data collection
- knowledge collection

Information Gathering - Consumer

- sources:
  - formal resources
- data collection
- knowledge collection

Information Gathering
Practice Activities

Phase 3 - Language Definition

Language Definition - Producer

- knowledge activities
  - Concept identification (single dataset)
  - UKC Alignment
- data activities
  - Dataset Filtering

Language Definition - Consumer

- knowledge activities
  - Concept identification (Purpose specific)
  - UKC Alignment
- data activities
  - Dataset Filtering

Language Definition
Practice Activities

Phase 4 - Knowledge Definition

Knowledge Definition - Producer

- knowledge activities
  - Teleology (single dataset)
  - Teleontology (single dataset)
- data activities
  - Dataset Cleaning Formatting

Knowledge Definition - Consumer

- knowledge activities
  - Teleology (purpose specific)
  - Teleontology (purpose specific)
- data activities
  - Dataset Cleaning Formatting

Knowledge Definition
Practice Activities

Phase 5 - Data Definition

Data Definition - Producer

- KG generation (for each single dataset)

Data Definition - Consumer

- KG generation (purpose specific)

Data Definition
Practice Activities

Evaluation

Knowledge Evaluation Metrics

Data Evaluation Metrics

Metadata definition

Metadata Purpose Definition

Metadata Information Gathering

Metadata Language Definition

Metadata Knowledge Definition

Metadata Data Definition

KG Query

GraphDB

SparQL

KGE Glossary

References

- Bella, Gábor, Alessio Zamboni, Fausto Giunchiglia, et al. (2016). "Domai based sense disambiguation in multilingual structured data". In: Diversity Workshop, ECAI.
- Giunchiglia, Fausto, Daqian Shi, et al. (2021). "Property-based Entity Type Graph Matching". In: *CEUR WORKSHOP PROCEEDINGS*. Vol. 3063. OM Workshop, ISWC 2021, pp. 1–12.
- Giunchiglia, Fausto, Alessio Zamboni, et al. (2021). "Stratified Data Integration". In: *2nd Int. Wshop On Knowledge Graph Construction (KGCW), ESWC*.