universität innsbruck

STI · INNSBRUCK

Knowledge Graph Lifecycle: Building and Maintaining Knowledge Graphs

Umutcan Şimşek, Kevin Angele, Elias Kärle, Juliette Opdenplatz, Dennis Sommer, Jürgen Umbrich, Dieter Fensel

Outline

1. Introduction
2. Knowledge Graph Lifecycle
3. Lessons learned
4. Conclusion and Future Work

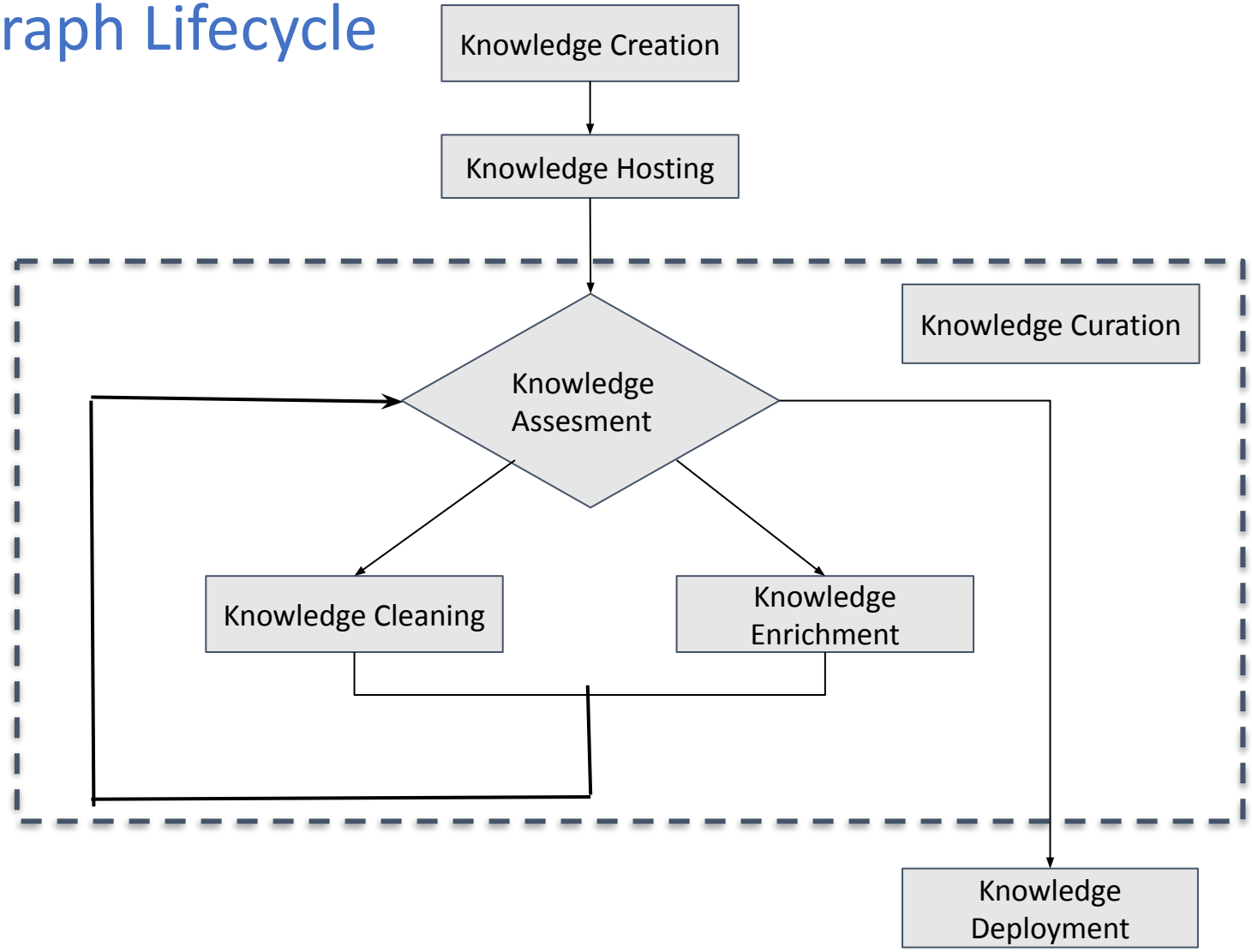
1. Introduction

- Knowledge graph lifecycle comes with two major challenges
  - map and integrate heterogeneous sources
  - maintain them to make a high-quality resource for an application at hand

- An experience report of our attempt to implement the lifecycle in various scenarios

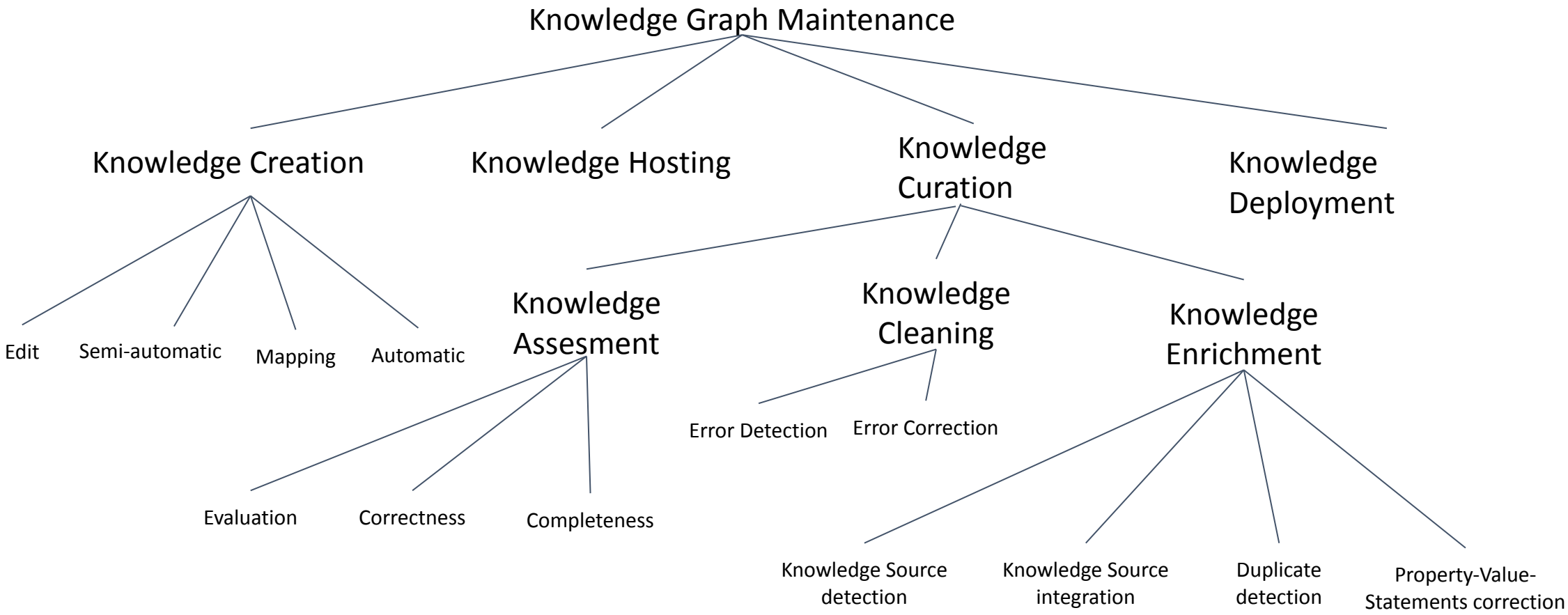
- Discussing lessons learned for
  - general implementation of the lifecycle
  - knowledge creation from heterogeneous sources (technical, conceptual and social challenges)
  - knowledge curation with different perspectives and scalability issues

2. Knowledge Graph Lifecycle

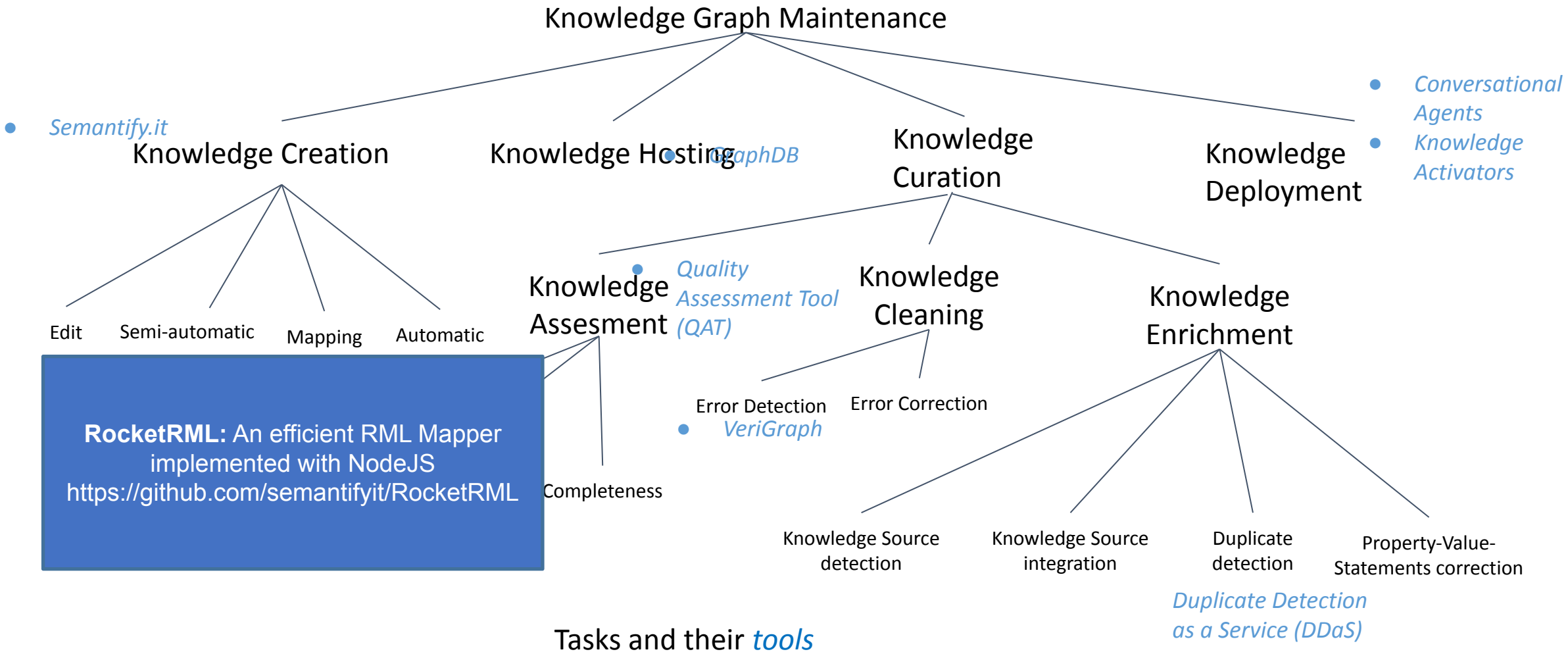


2. Knowledge Graph Lifecycle

Tasks

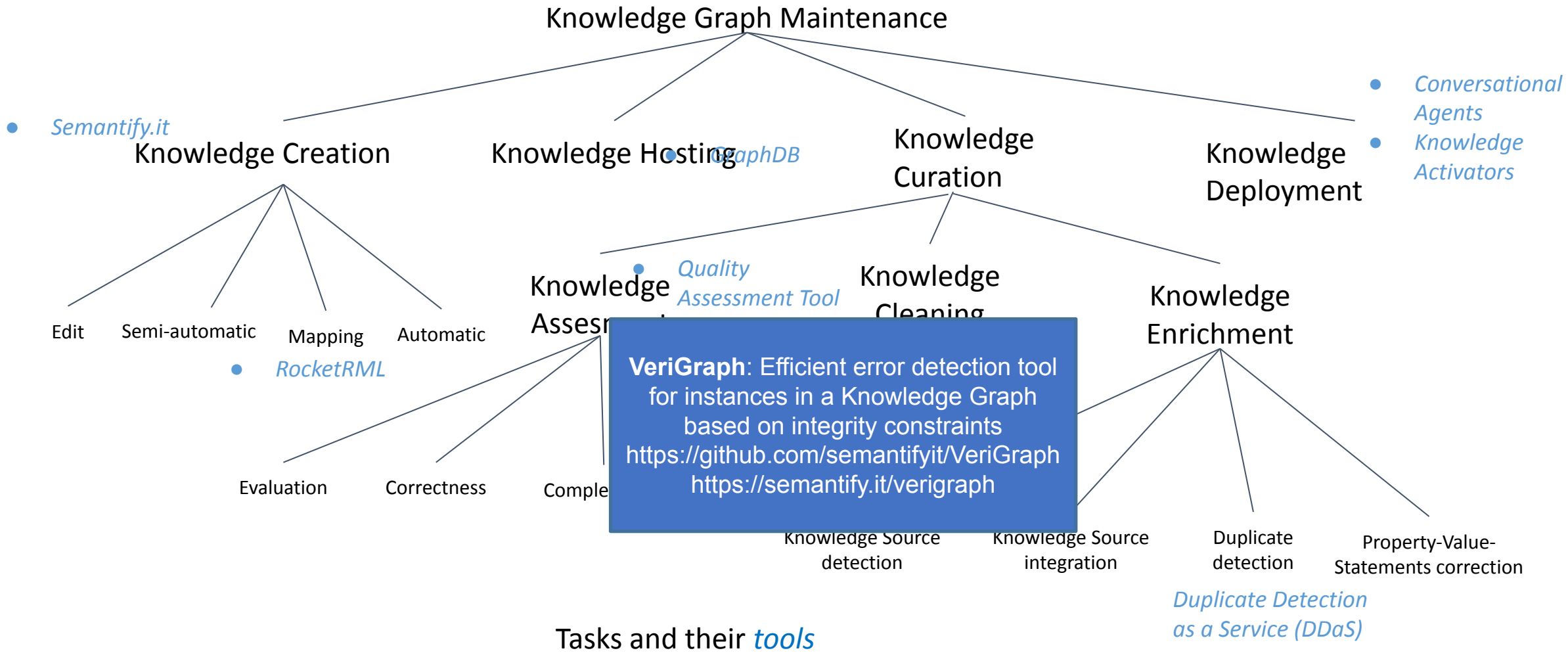


2. Knowledge Graph Lifecycle



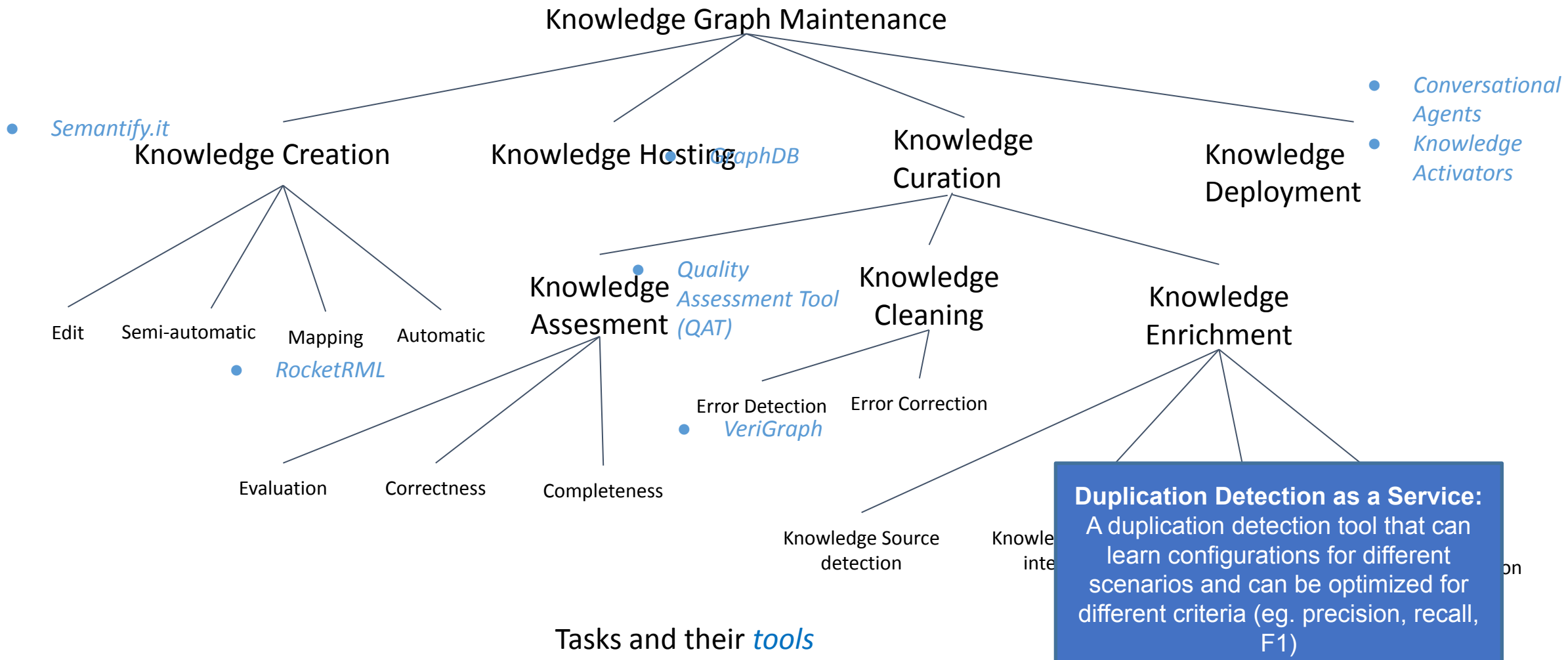
Tasks and their *tools*

2. Knowledge Graph Lifecycle



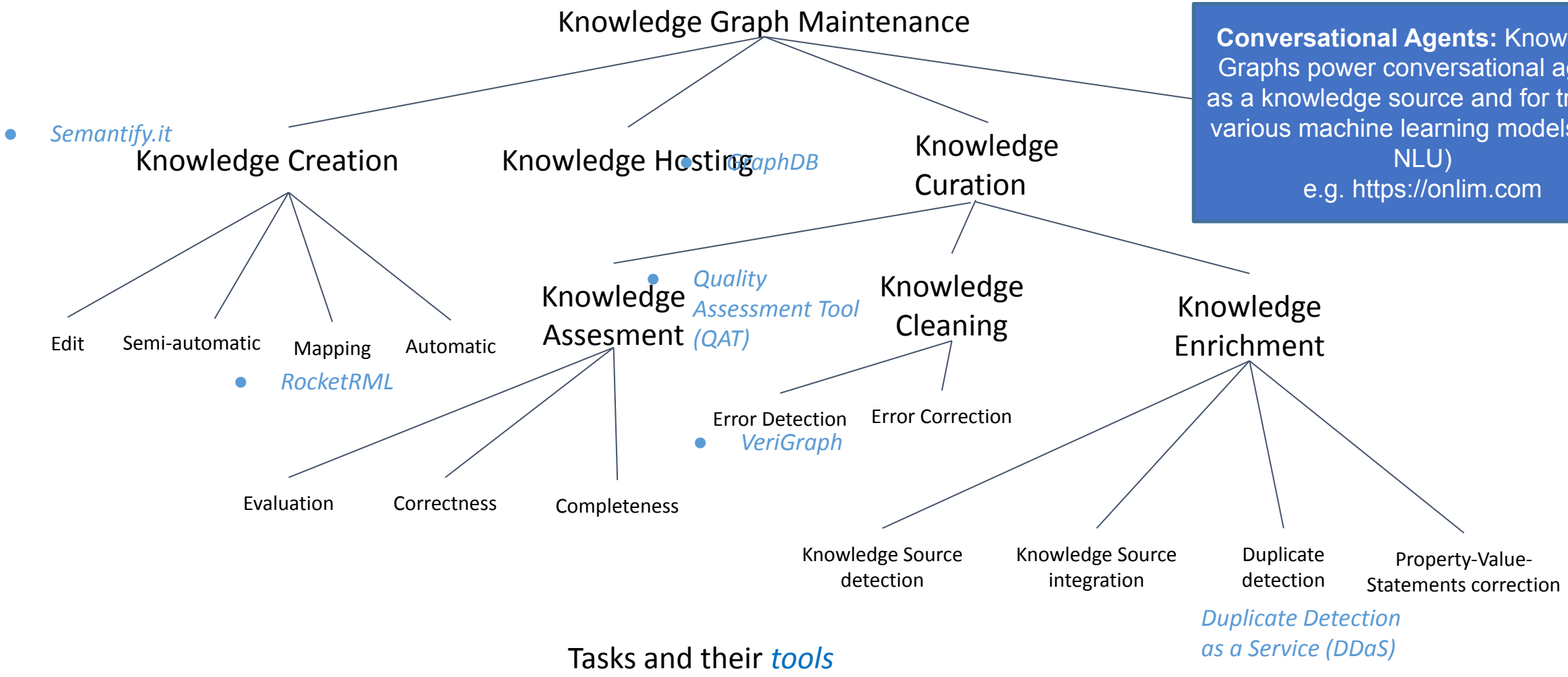
Tasks and their *tools*

2. Knowledge Graph Lifecycle



Tasks and their *tools*

2. Knowledge Graph Lifecycle



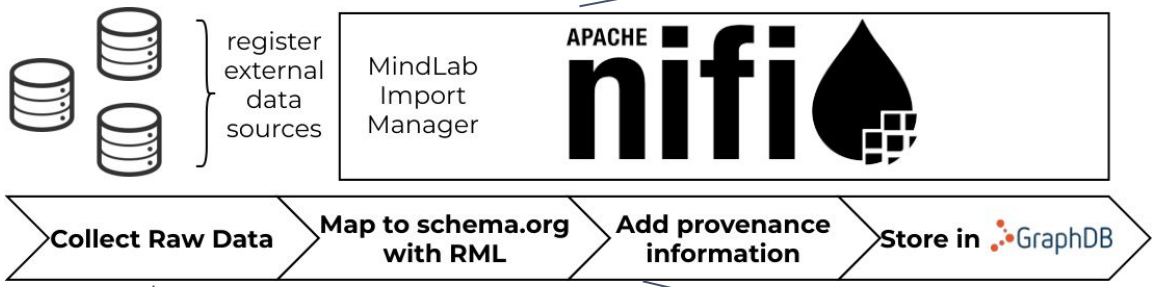
Conversational Agents: Knowledge Graphs power conversational agents as a knowledge source and for training various machine learning models (e.g. NLU)
e.g. https://onlim.com

Tasks and their *tools*

2. Knowledge Graph Lifecycle

Knowledge Generation

highly configurable ETL tool to orchestrate the pipeline



PROV-O and schema.org used

Typically in JSON or XML format

- Domain knowledge shared with domain-specific patterns of schema.org (and its extensions)
- YARRML syntax was an important decision factor for RML

```
#prefixes
sources:
...
mappings:
  acc:
    sources:
      - acc
    s: ml:$(@Id)
    po:
      - [a, {function: myfunc:getType, parameters:
      ↪  ["$(Details/Topics/Topic/@Id)"]}]
      - [schema:name, "$(Details/Names/Translation[@Language='de'])",
      ↪  de~lang]
      - [schema:name, "$(Details/Names/Translation[@Language='en'])",
      ↪  en~lang]
      ...
      - [schema:openingHoursSpecification, {mapping: hours, join: [@Id,
      ↪  ../../../../@Id]}]
  ...
```


<- return to DS List
show SHACL serialization

LocalBusiness

A particular physical business or branch of an organization. Examples of LocalBusiness include a restaurant, a particular branch of a restaurant chain, a branch of a bank, a medical practice, a club, a bowling alley, etc.

External link External link to schema.org

Property	Expected Type	Description	Cardinality
address	PostalAddress	Physical address of the item.	1
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.	0..1
contactPoint	ContactPoint	A contact point for a person or organization.	0..1
department	Organization	A relationship between an organization and a department of that organization, also described as an organization (allowing different urls, logos, opening hours). For example: a store with a pharmacy, or a bakery with a cafe.	0..1
description	Text	A description of the item.	1
email	Text	Email address.	1
faxNumber	Text	The fax number.	0..1
founder	Person	A person who founded this organization.	0..1
foundingDate	Date	The date that this organization was founded.	0..1
geo	GeoCoordinates	The geo coordinates of the place.	0..1
hasMap	URL	A URL to a map of the place.	0..1
hasOfferCatalog	OfferCatalog	Indicates an OfferCatalog listing for this Organization, Person, or Service.	0..1
image	URL	An image of the item. This can be a URL or a fully described ImageObject.	1

2. Knowledge Graph Lifecycle

Knowledge Cleaning

- Error detection with integrity constraints defined based on domain-specific patterns (implemented with SHACL)

- VeriGraph developed
  - supports a subset of SHACL

- some shortcomings encountered in the existing SHACL verifiers
  - tied to a triple store (e.g. Stardog)
  - does not work directly on SPARQL endpoints (e.g. reference implementation from TopBraid)
  - do not finish verification when the knowledge graph size go over magnitude of millions (e.g. RDFUnit)
    - the reason is most probably the network overhead of running SPARQL queries and creating verification reports

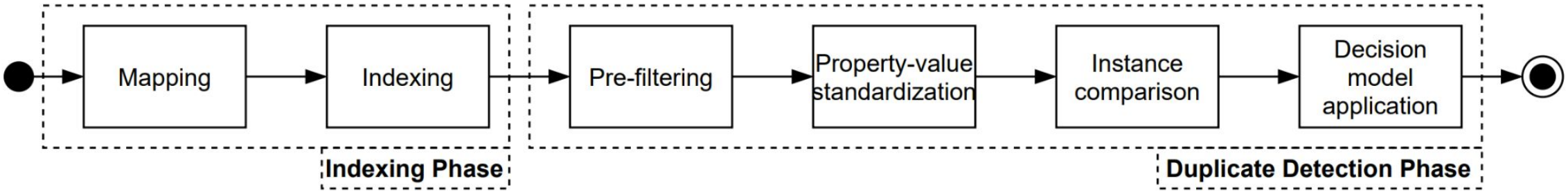
- Verification per instance: The instance relevant to a pattern is retrieved with a single SPARQL query (similar to using DESCRIBE queries). Less overhead over the network.

2. Knowledge Graph Lifecycle

Knowledge Enrichment

Duplicate Detection as a Service (**DDaaS**) is a service-oriented framework that allows linking duplicate instances within a Knowledge Graph or among Knowledge Graphs.

Inspired by DUKE, LIMES and SILK



DDaaS allows configuration learning for also indexing and pre-filtering phases and allows optimization for Precision, Recall or F-Score

Most tools allow configuration learning for only duplication detection phase and optimizes F-score.

2. Knowledge Graph Lifecycle

Knowledge Deployment

- Open Touristic Knowledge Graphs
  - **Tyrolean Tourism Knowledge Graph**: Integrates heterogenous data from 10+ Destination Management Organizations (DMO) - 12B+ triples. University prototype and showcase.
  - **German Tourism Knowledge Graph** - Integrates heterogenous data from the Regional Marketing Organizations (LMO*) in Germany. Real-world application

- Dialog systems
  - Conversational agents that help users to achieve their goals with the help of Knowledge Graphs:
    - Knowledge Graphs as a source of domain knowledge (about static, dynamic and active data)
    - Knowledge Graphs for training Natural Language Understanding models

* Landesmarketing Organization

2. Knowledge Graph Lifecycle

Knowledge Deployment

We build the **Tyrol Knowledge Graph (TKG)** as our first showcase

- It is a published in **GraphDB** providing a **SPARQL** endpoint for the provisioning of touristic data of Tyrol, Austria.
- The TKG currently contains data about touristic infrastructure like accommodation businesses, restaurants, points of interests, events, recipes, et.
- Currently the TKG contains over 12B+ statements, 55% are explicit and 45% are inferred.
- https://tirol.kg/

2. Knowledge Graph Lifecycle

Knowledge Deployment

- The **German Tourism Knowledge Graph** will integrate semantically annotated tourism-related data from 16 LMOs/Magic Cities.

- Contracted by the Deutsche Zentrale für Tourismus (DZT), planned to be finished in 2022.
- An ecosystem for Knowledge Creation, Hosting and Deployment is being built.

- Compliant with the domain-specific patterns developed by **The Open Data Tourism Alliance (ODTA).**
  - https://ds.sti2.org

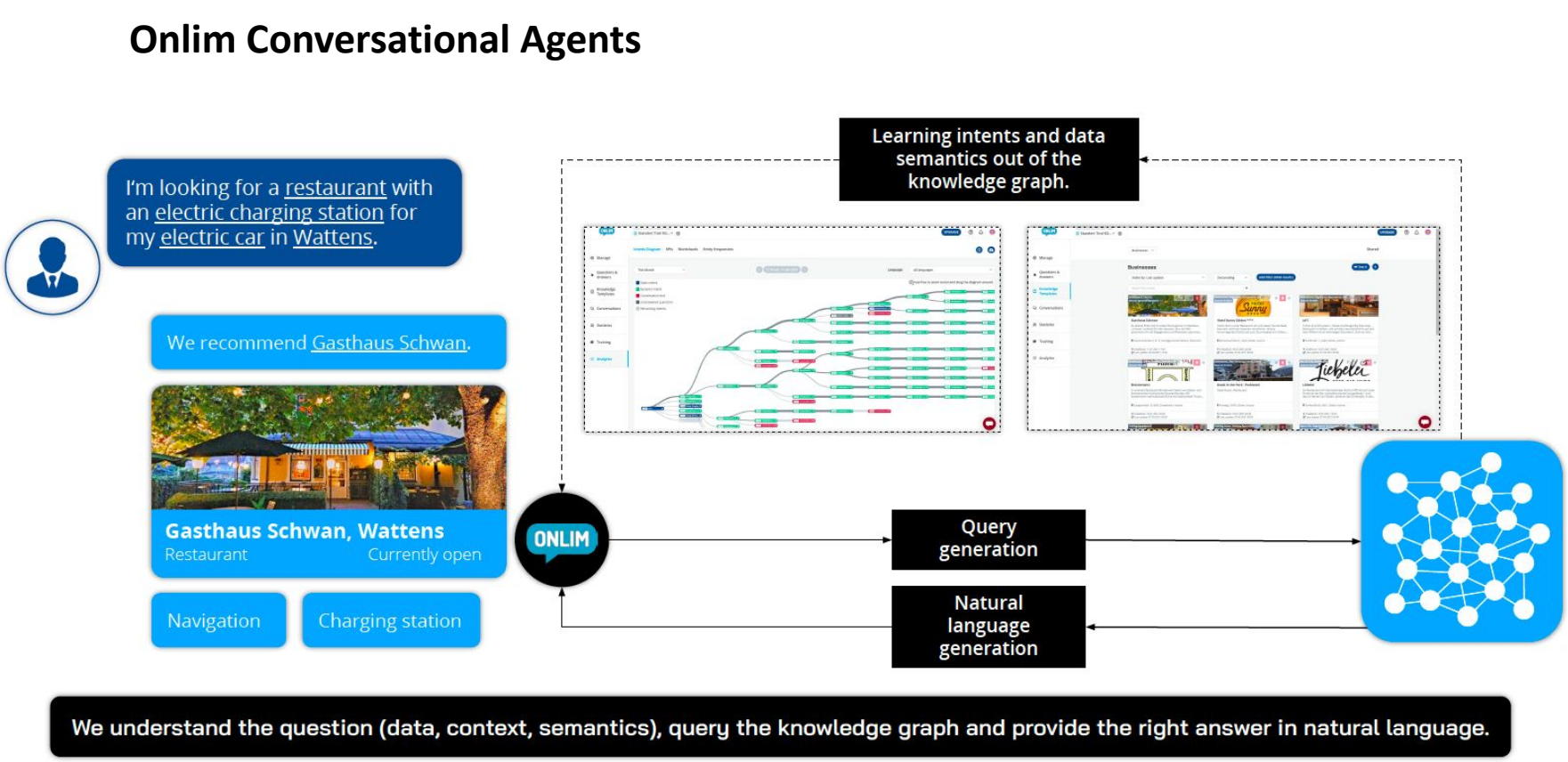
- Reference real-world implementation of the full lifecycle

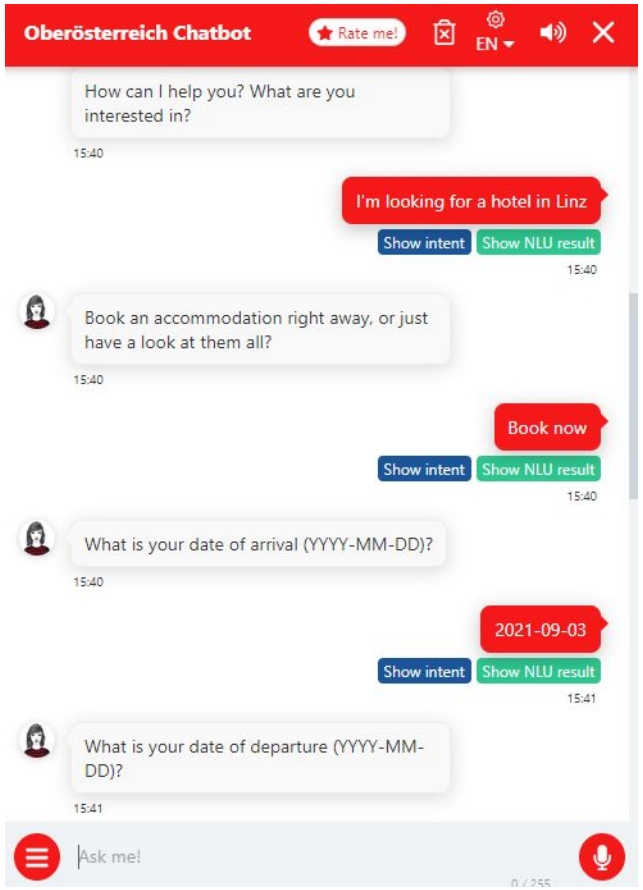
- Read more at:
  - https://open-data-germany.org/open-data-germany/

2. Knowledge Graph Lifecycle

Knowledge Deployment

Onlim Conversational Agents





3. Lessons Learned

General Lessons

- **Orchestration of different tasks in the lifecycle**
  - Many of the individual tasks have been addressed in academia and industry, however a tool to orchestrate the lifecycle lacks
  - Need for open APIs and increased interoperability between tools supporting different tasks in the lifecycle

- **Community effort needed to maintain existing research products**
  - There is a great amount of important work in the academia addressing various tasks in the lifecycle, however many of them are not maintained (documentation/code outdated)
  - Is an effort similar to Apache Software Foundation a solution?

3. Lessons Learned

Knowledge Creation

- **Real data is not perfect, knowledge creation is not trivial**
  - In TKG use case we realized that it is very common to receive data without any fields for join operations
  - We currently work around this by using JSONPath+ and introducing a new construct called PATH~ to join based on the exact path of nested elements

- **Conceptual and social challenges stand**
  - Declarative mappings solve a technical problem
  - The domain experts must define the domain by identifying relevant types, properties, and constraints and communicate them to the developers and mapping creators. This is particularly challenging when these actors are distributed across different organizations, as it is in the German Tourism Knowledge Graph.
  - Simple human- and machine-understandable patterns ofschema.org and its extensions published by domain experts improve the knowledge creation process significantly.

3. Lessons Learned

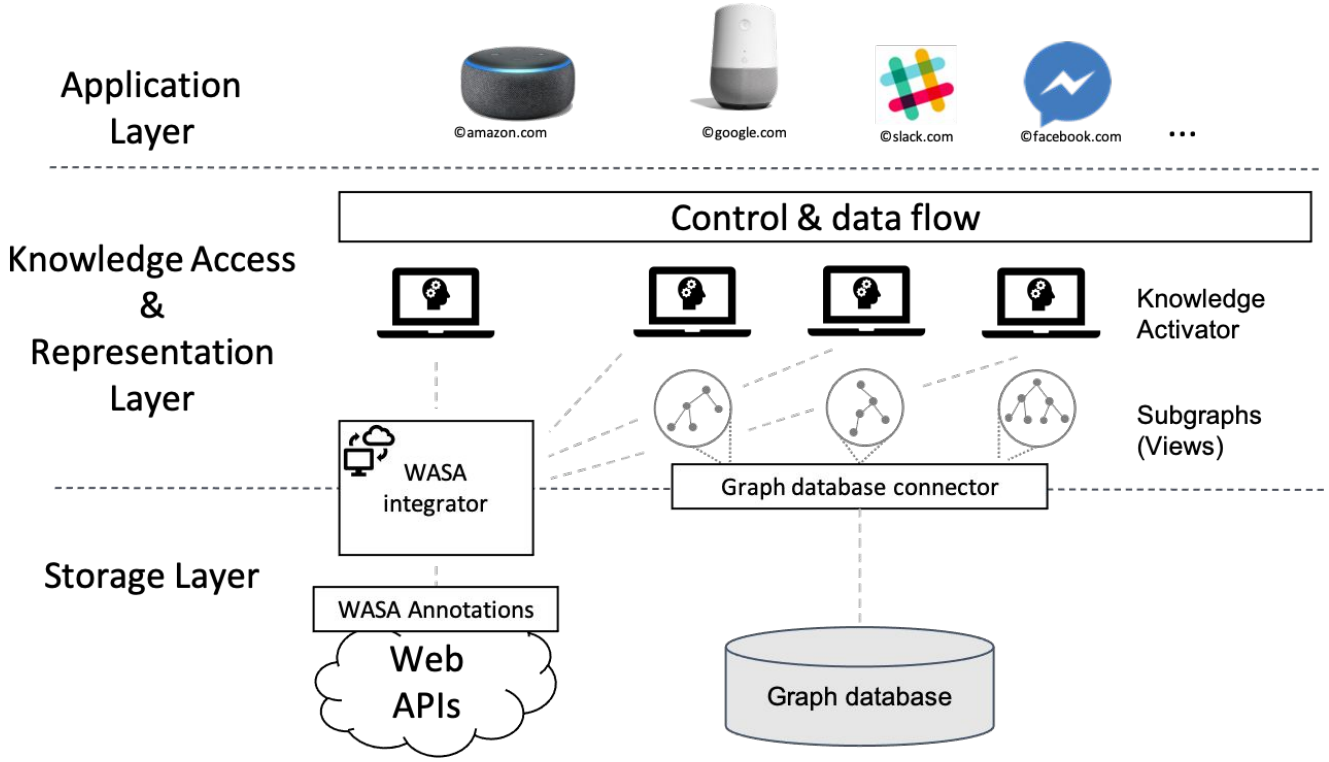
Knowledge Curation

- **Different perspectives on knowledge integrity**
  - One experience we had with Tyrolean and German Tourism Knowledge Graph use cases is that the different instances of the same type may have different expected shapes.
    - For instance, a generic Organization shape may require schema:vatID property however for a schema:Organization instance that is the value of organizer property of an event only the name property may be interesting. A SHACL shape that targets the schema:Organization type would verify both Organization instances, which is not the intended behavior.
    - Domain-specific patterns as types with local properties and ranges and directly connect to instances

- **Distinguishing between different kind of constraints may help optimizing cleaning process**
  - Verifying mappings vs verifying the knowledge graph
  - For the cases where mappings can be controlled
  - Important for improving verification efficiency for time-sensitive applications like conversational agents

4. Conclusion and Future Work

- Introduced the knowledge graph lifecycle and solutions produced for some of the tasks
- Discussed lessons-learned
- Time-sensitive applications like conversational agents require high-quality data really fast.
  - this may be challenging as the knowledge graph grows
- Part of our current and future work is to only curate a relevant part of a knowledge graph for an application at a given time. This allows
  - Supporting different point of views in “micro TBoxes”
  - Reducing the time needed for curation



www.sti2.at

Twitter: @umutsims

www.uibk.ac.at