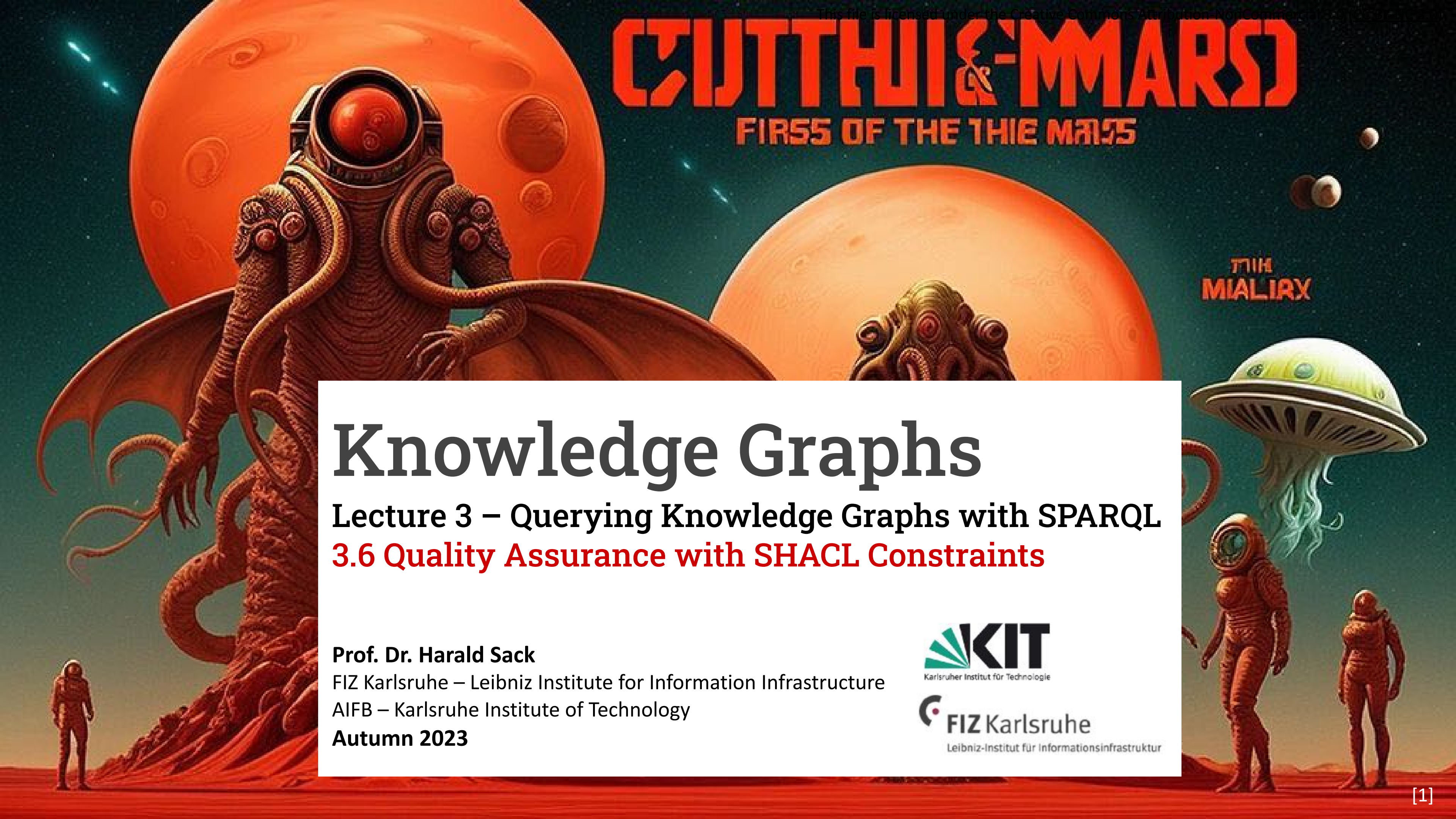


Knowledge Graphs

Lecture 3 – Querying Knowledge Graphs with SPARQL

3.6 Quality Assurance with SHACL Constraints

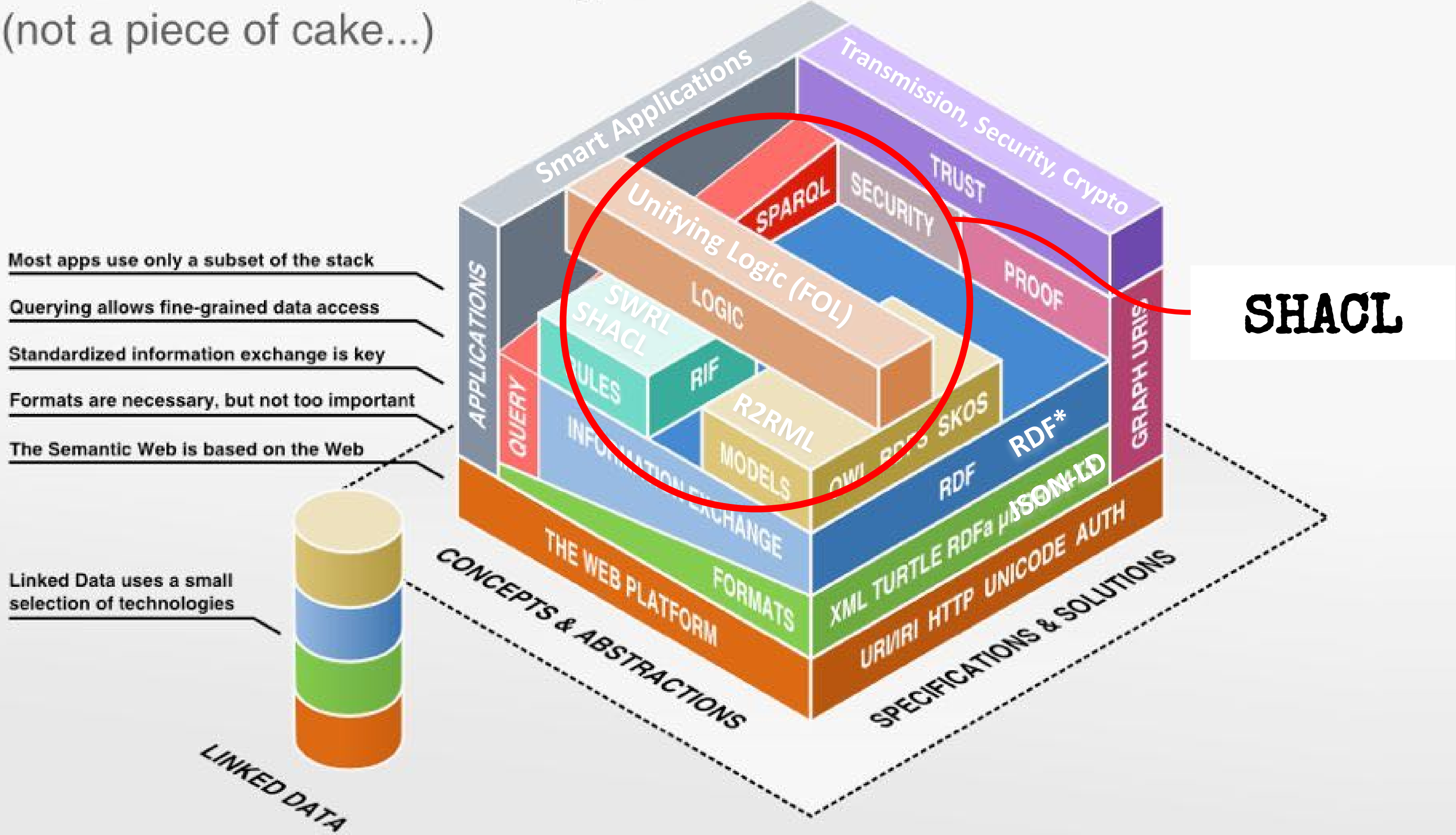
Prof. Dr. Harald Sack
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure
AIFB – Karlsruhe Institute of Technology
Autumn 2023

[1]

Knowledge Graphs

Lecture 3: Querying Knowledge Graphs with SPARQL

The Semantic Web Technology Stack
(not a piece of cake...)



[2]

Open & Closed World Assumption – OWA vs CWA

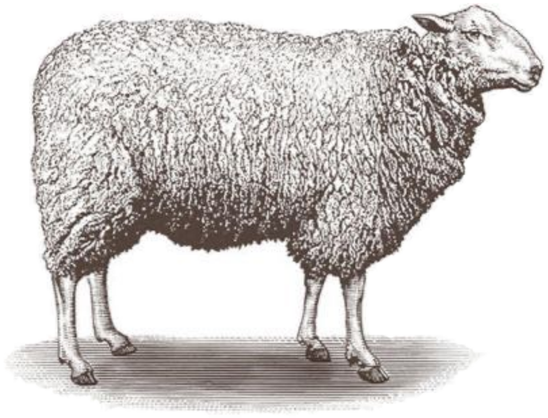
`A sheep is an animal with four legs.`

- Answer under CWA Assumption:
- Answer under OWA assumption:

Question: Can sheep fly?

No, sheep can't fly.

No idea, but probably yes (according to our knowledge base).



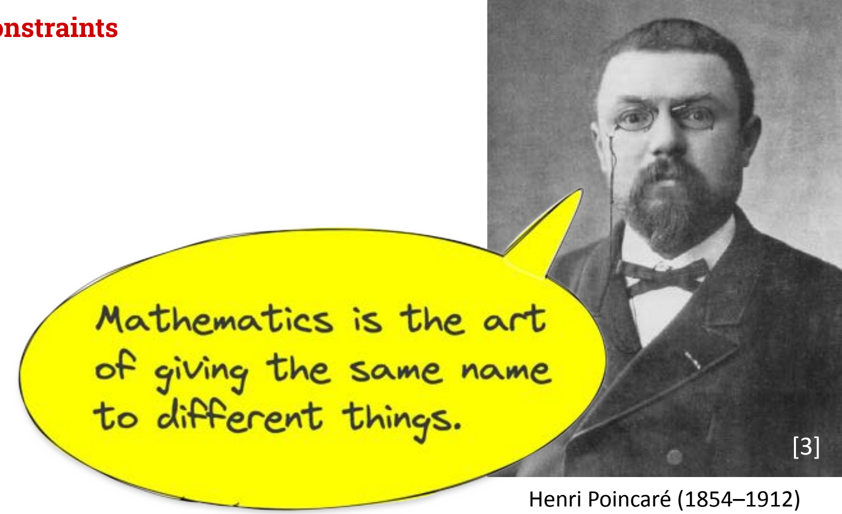
- In the OWA, unless we have a statement (or we can infer) "*sheep can/cannot fly*" we return "**don't know**".
- In the real world, we are used to deal with incomplete information.

Open & Closed World Assumption – OWA vs CWA

- In the Semantic Web we expect people to extend our own models (*but we don't worry in advance how*).
- The OWA assumes incomplete information by default.
- Therefore, we can **intentionally underspecify our models and allow others to reuse and extend.**

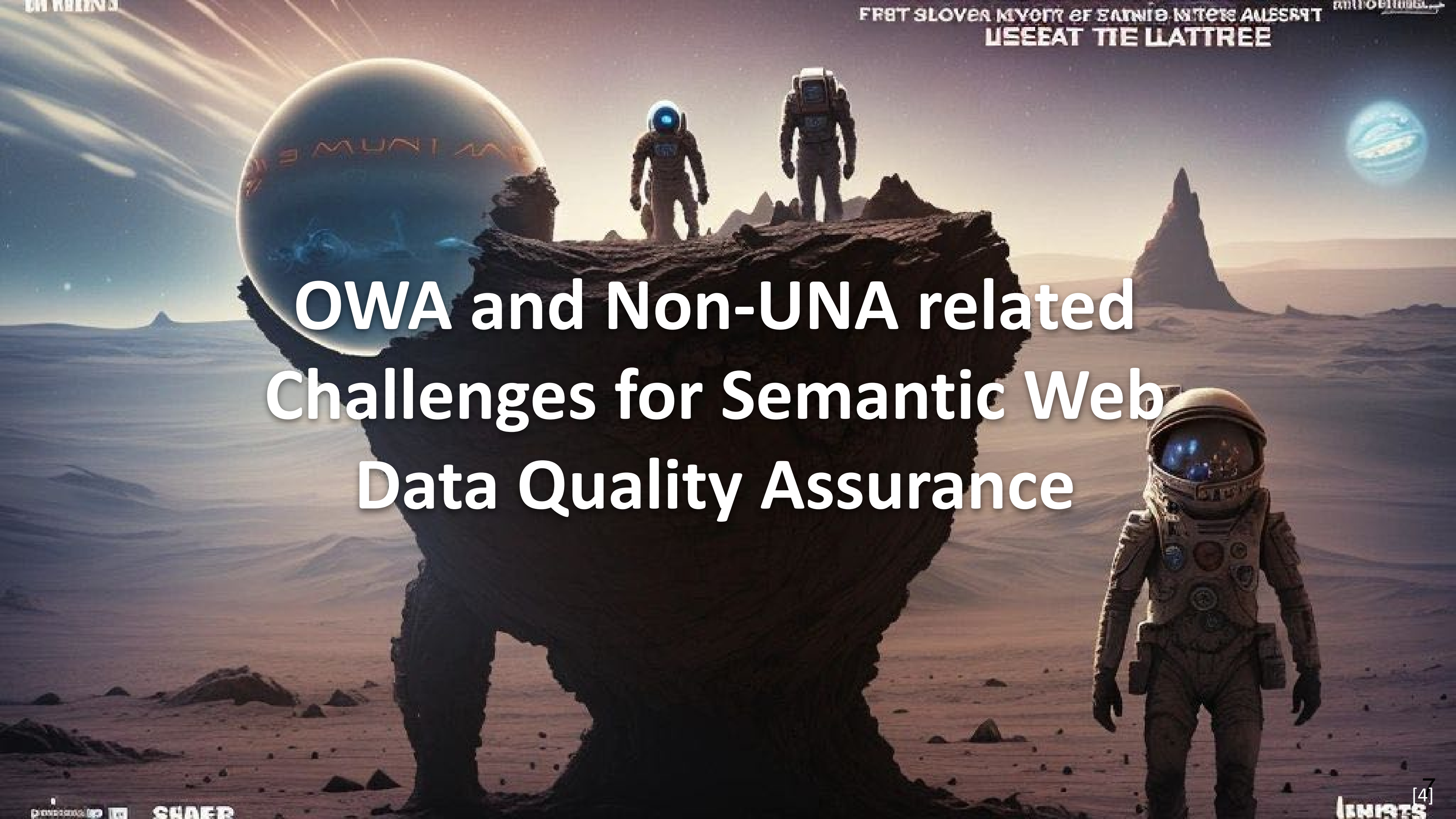
```
A sheep is an animal with four legs that
(if lifted) can fly.
```

Unique Name Assumption – UNA



Henri Poincaré (1854–1912)

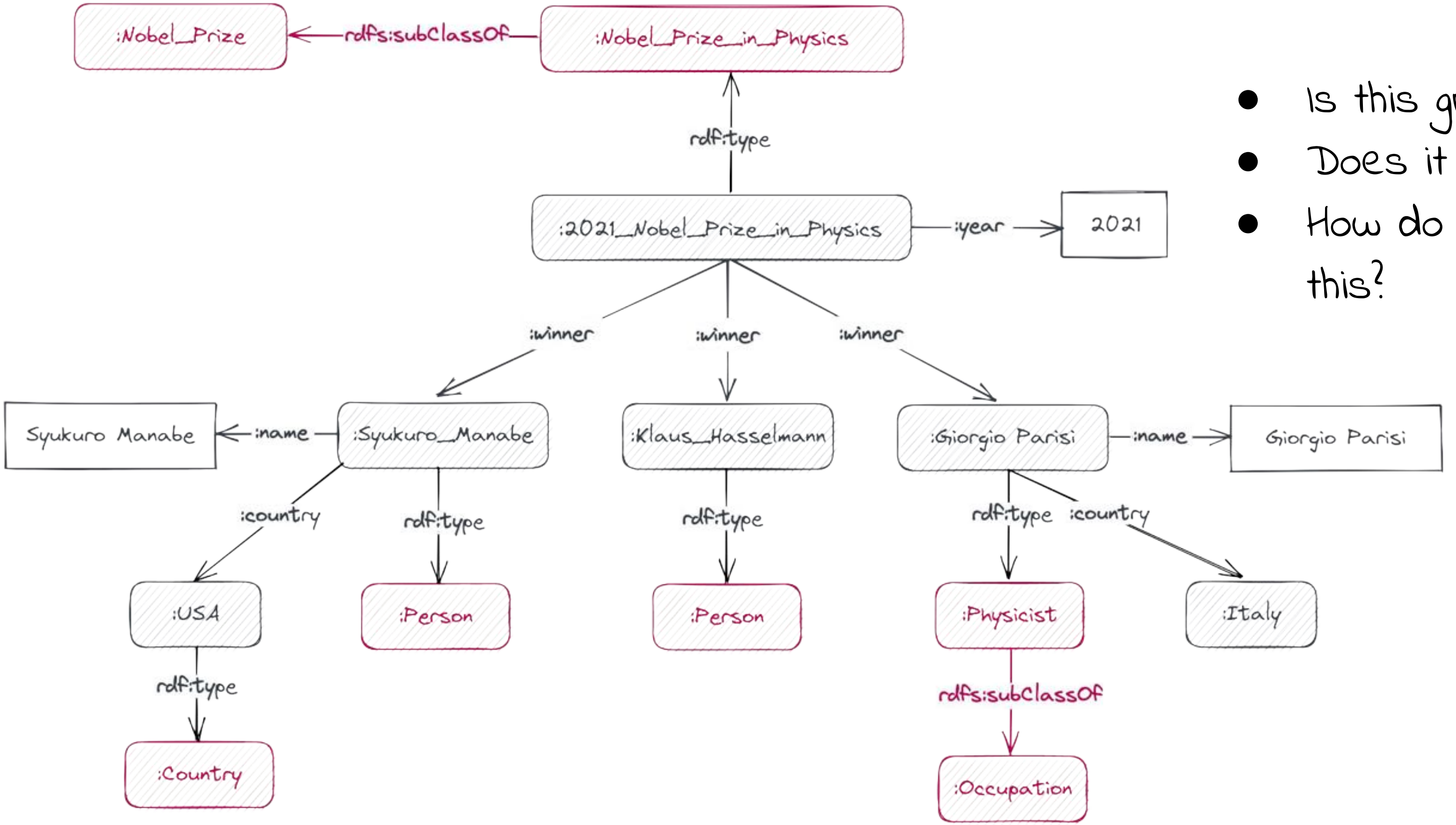
- In logics with **UNA**, **different names always refer to different entities in the world**.
- **OWL** does NOT support UNA (because of OWA).
- Consequences for the Semantic Web:
    - Different entities have to be **declared to be different** (otherwise they are potentially identical). `owl:differentFrom`, `owl:disjointWith`
    - Identical entities also have to be **declared to be identical** (otherwise they are potentially different). `owl:sameAs`, `owl:equivalentClass`



OWA and Non-UNA related Challenges for Semantic Web Data Quality Assurance

[4]

Graph Data – Validation

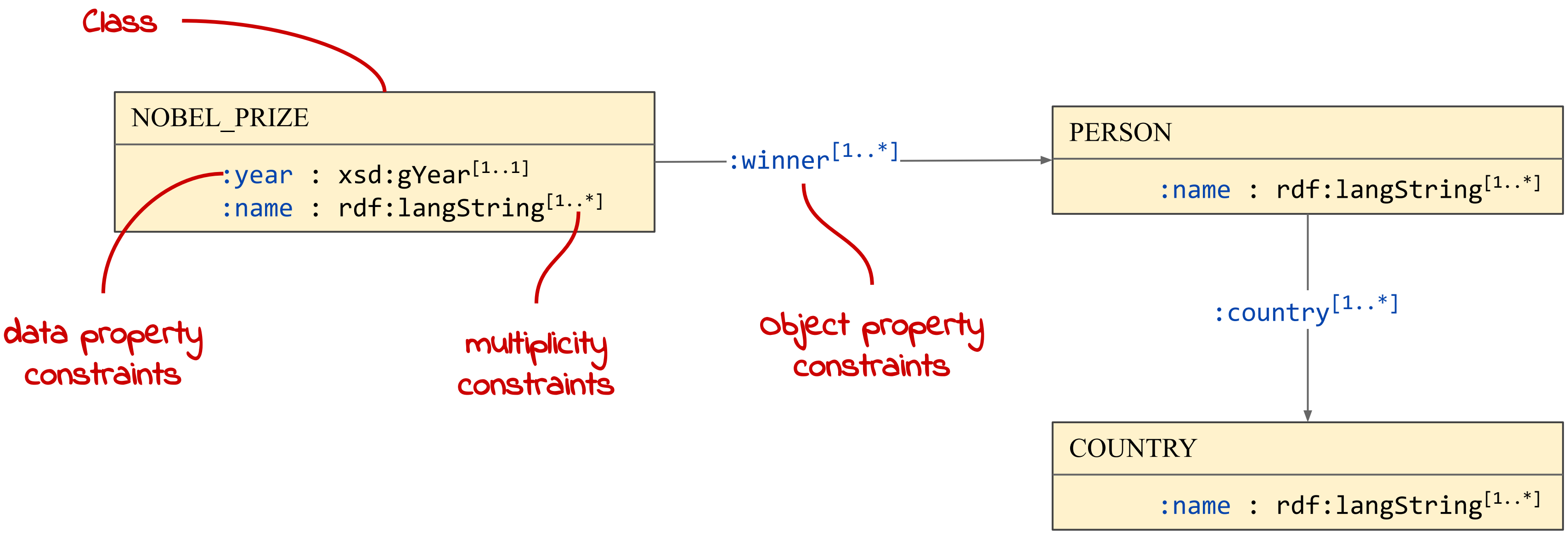


- Is this graph complete?
- Does it contain errors?
- How do we measure this?

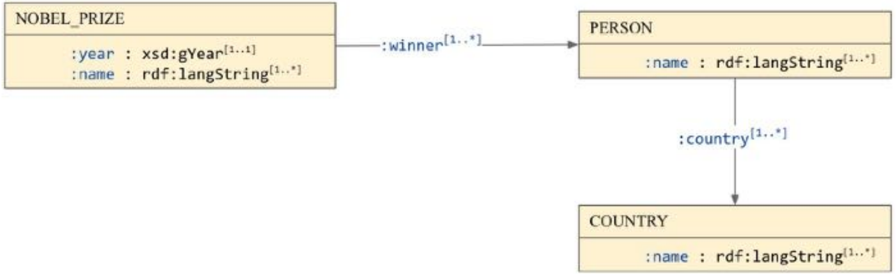
Shapes

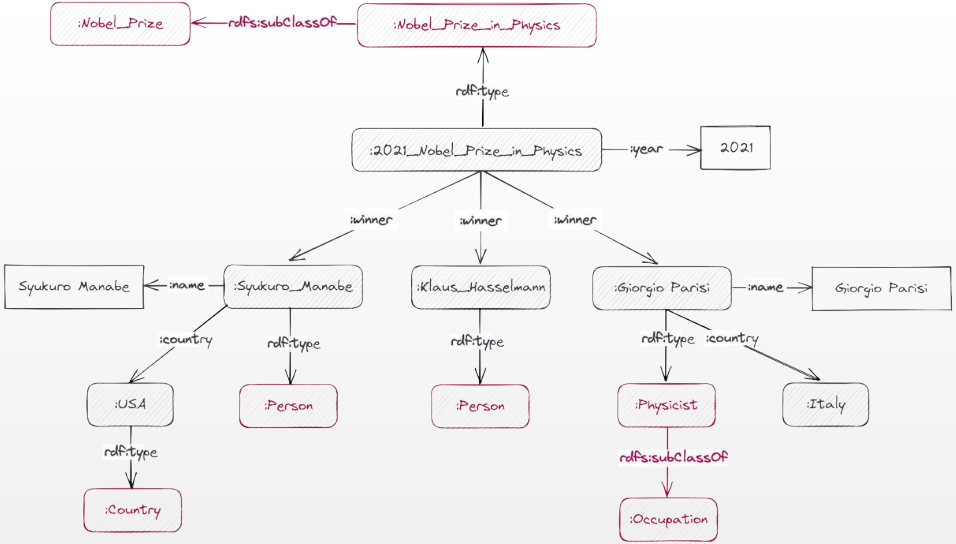
[4]

Shapes Graph – Validation Schema



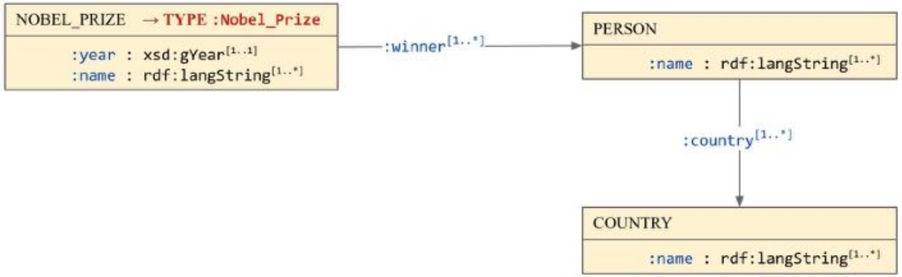
Validate RDF Graphs

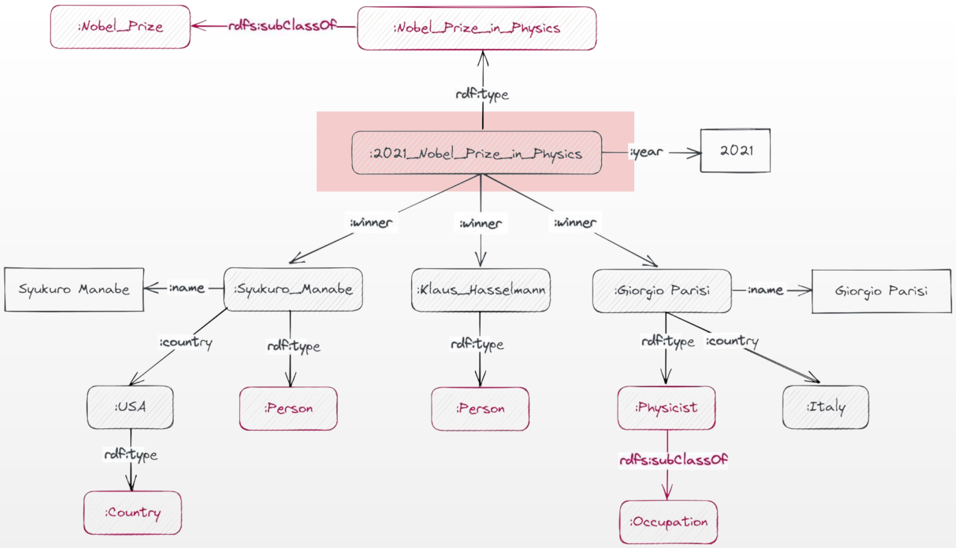




- Does the graph pass our schema?
- We have not yet defined a **target** for a shape, so we don't know which shape applies to which node in the data.

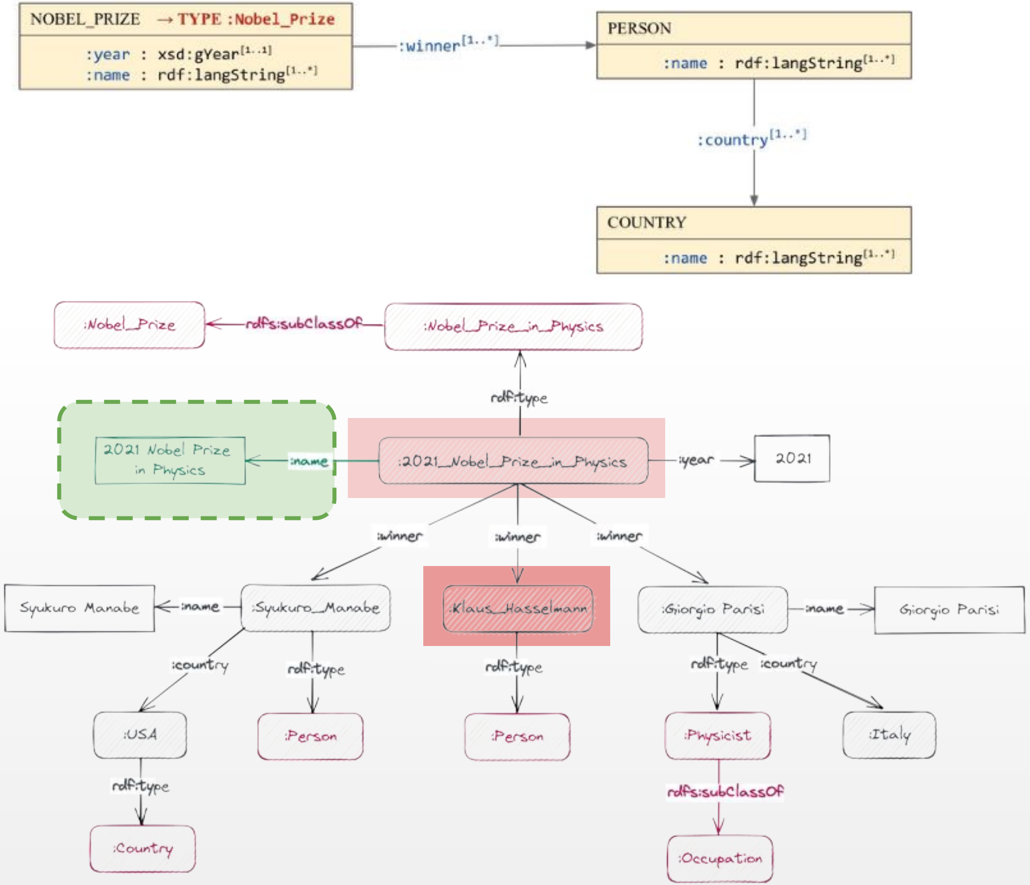
Shapes Graph – Define a Target





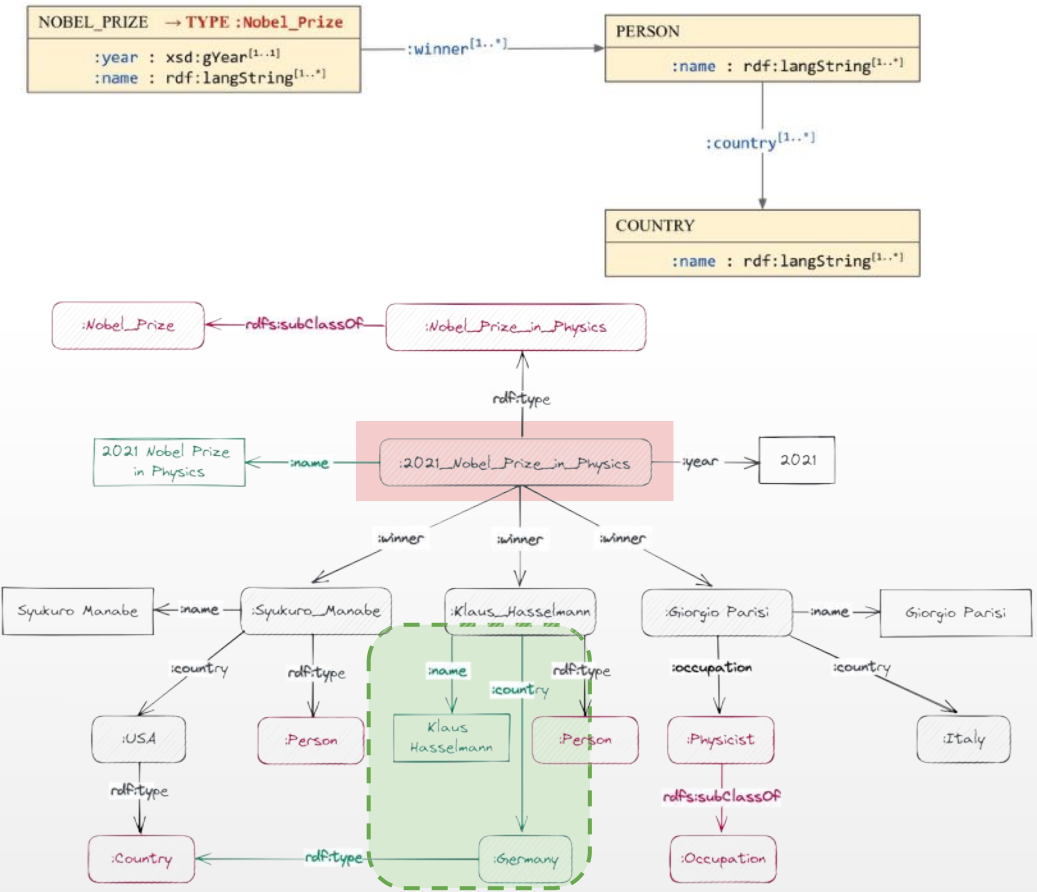
- Does the graph pass our schema?
- No, `:2021_Nobel_Prize_in_Physics` does not satisfy NOBEL_PRIZE
- We are missing a `:name` for `:2021_Nobel_Prize_in_Physics`

Shapes Graph – Validation



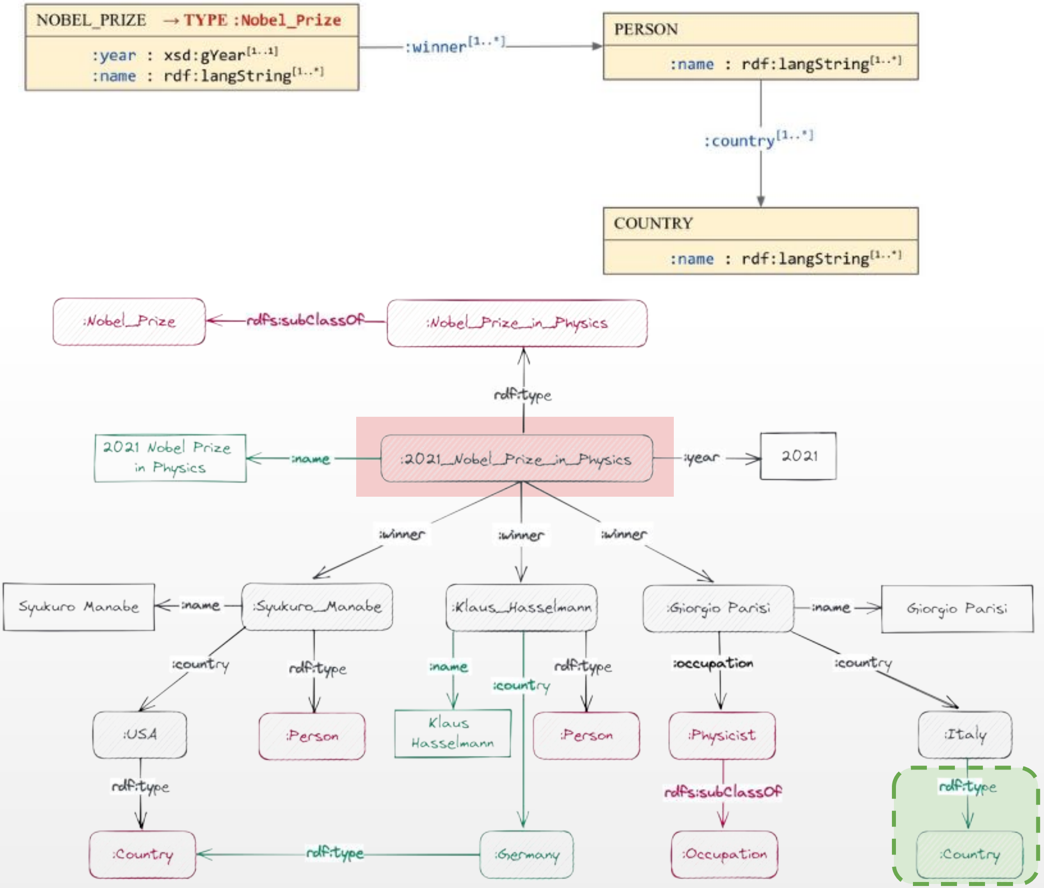
- What about now?
- No, any winner of `:2021_Nobel_Prize_in_Physics` still has to satisfy PERSON
- We are missing `:name` and `:country` for `:Klaus_Hasselmann`

Shapes Graph – Validation



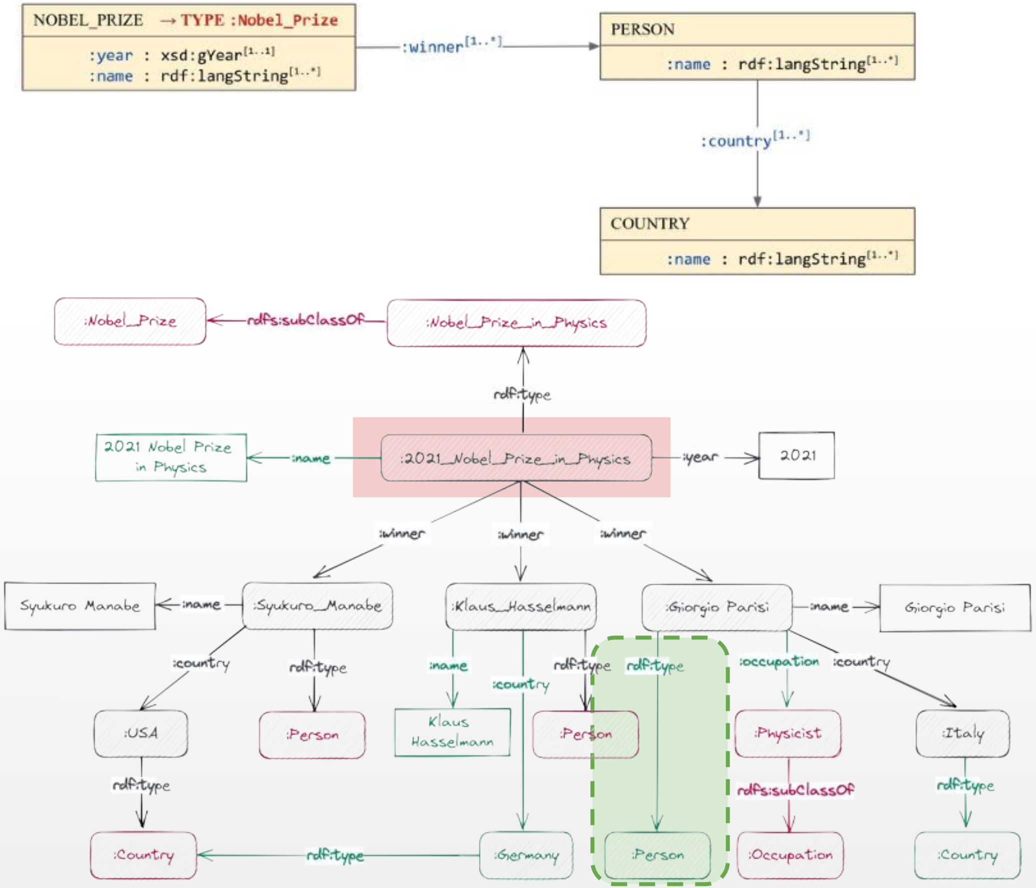
- What about now?
- No, any winner of `:2021_Nobel_Prize_in_Physics` still has to satisfy PERSON
- For `:Giorgio_Parisi` PERSON requires that COUNTRY is satisfied

Shapes Graph – Validation



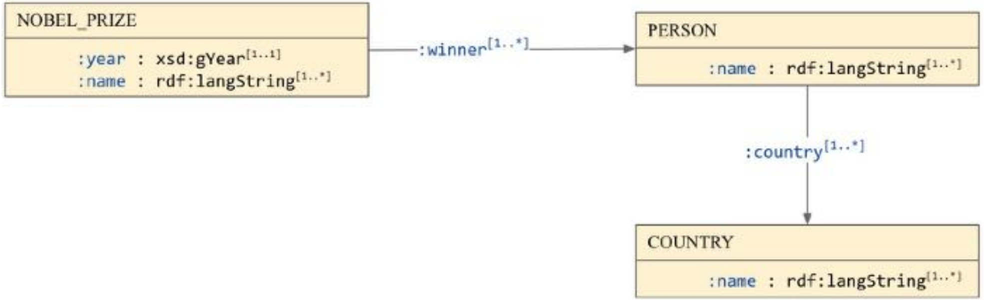
- What about now?
- No, any winner of `:2021_Nobel_Prize_in_Physics` still has to satisfy PERSON
- `:Giorgio_Parisi` is a :Physicist and not a :Person

Shapes Graph – Validation



- And now?
- Yes (for target :Nobel_Prize)
- However, for quality assurance of the entire graph we have to continue the validation.

Shapes vs. OWL



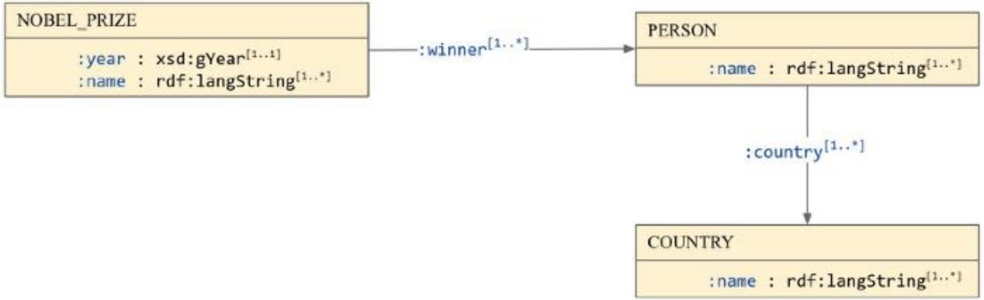

```
:Nobel_Prize rdfs:subClassOf
 [owl:allValuesFrom xsd:gYear ; owl:onProperty :year],
 [owl:cardinality 1 ; owl:onProperty :year],
 [owl:allValuesFrom rdf:langstring ; owl:onProperty :name],
 [owl:minCardinality 1 ; owl:onProperty :name],
 [owl:allValuesFrom :Person ; owl:onProperty :winner] .

:Person rdfs:subClassOf
 [owl:allValuesFrom rdf:langstring ; owl:onProperty :name],
 [owl:minCardinality 1 ; owl:onProperty :name],
 [owl:allValuesFrom :Country ; owl:onProperty :country] .

:Country rdfs:subClassOf
 [owl:allValuesFrom rdf:langstring ; owl:onProperty :name],
 [owl:minCardinality 1 ; owl:onProperty :name] .
```

- OWL follows **Open World Assumption** (OWA) and does not support the **Unique Name Assumption** (UNA).
- Therefore it will be difficult to test for
  - completeness and
  - duplicates.

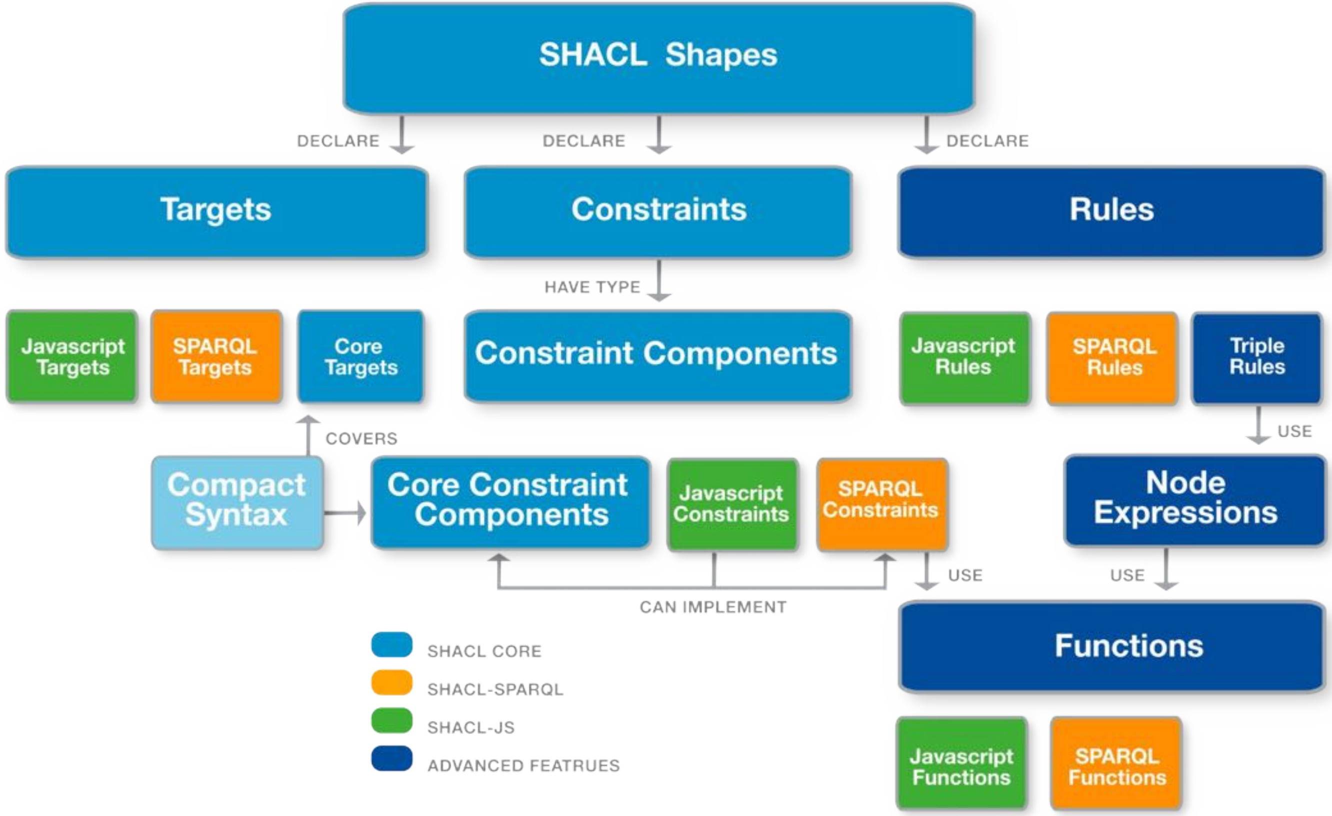
Shapes vs. SPARQL




```
SELECT DISTINCT ?award WHERE {
 ?award a :Nobel_Prize .
 OPTIONAL { ?award :year ?year1 .}
 OPTIONAL { ?award :year ?year2 .}
 FILTER(!bound(?year1)||datatype(?year1)!=xsd:gYear||?year1!=?year2)
 OPTIONAL { ?award :name ?name .}
 FILTER(!bound(?name)||lang(?name)=””)
 OPTIONAL {?award :winner :winner .}
#Check that ?winner satisfies PERSON shape
}
```

- SPARQL provides the correct semantics.
- However, constraints are difficult to phrase.

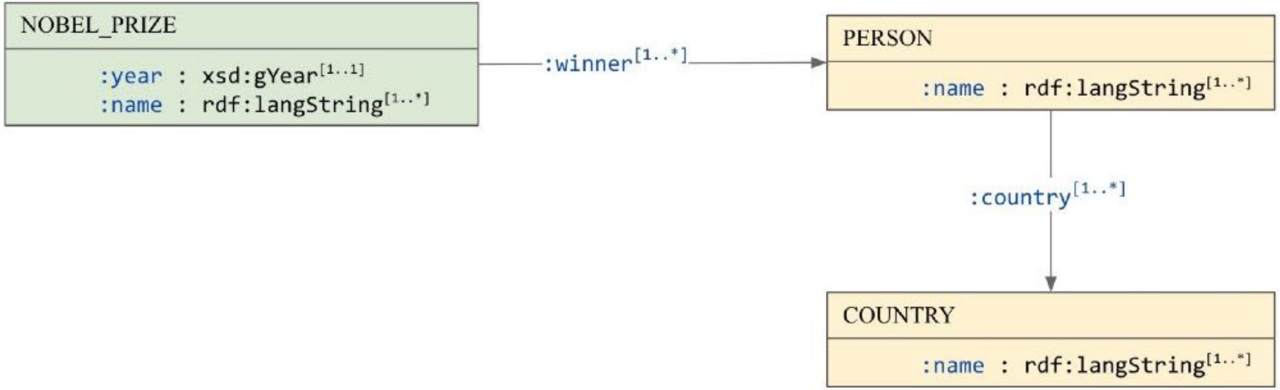
The Shape Constraint Language – SHACL



W3C SHACL Reference

https://www.topquadrant.com/wp-content/uploads/2017/09/SHACL_outlined-text-+-24.png

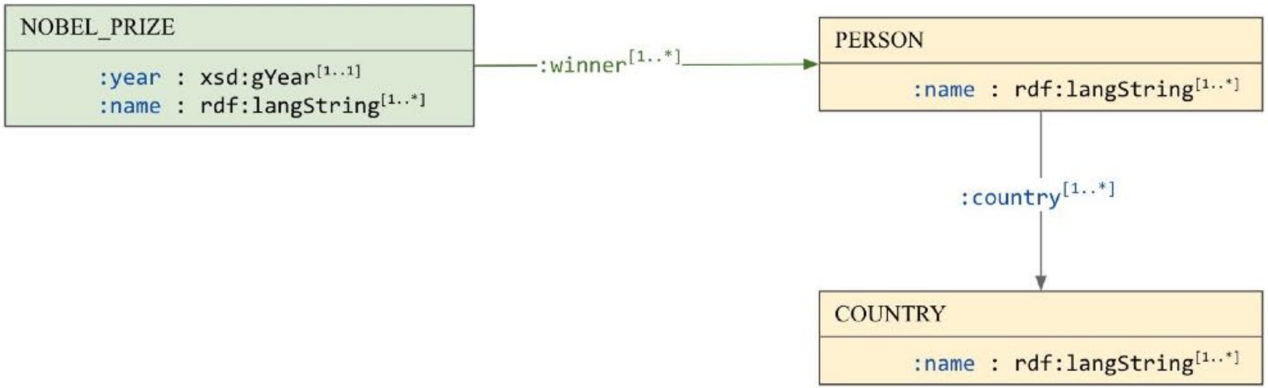
SHACL – Node Shapes and Property Shapes




```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix s: <http://example.org/shapes/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix : <http://example.org/> .

s:Nobel_Prize a sh:NodeShape ;
  sh:targetClass :Nobel_Prize ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :year ; sh:datatype xsd:gYear ; sh:minCount 1 ; sh:maxCount 1] .
```

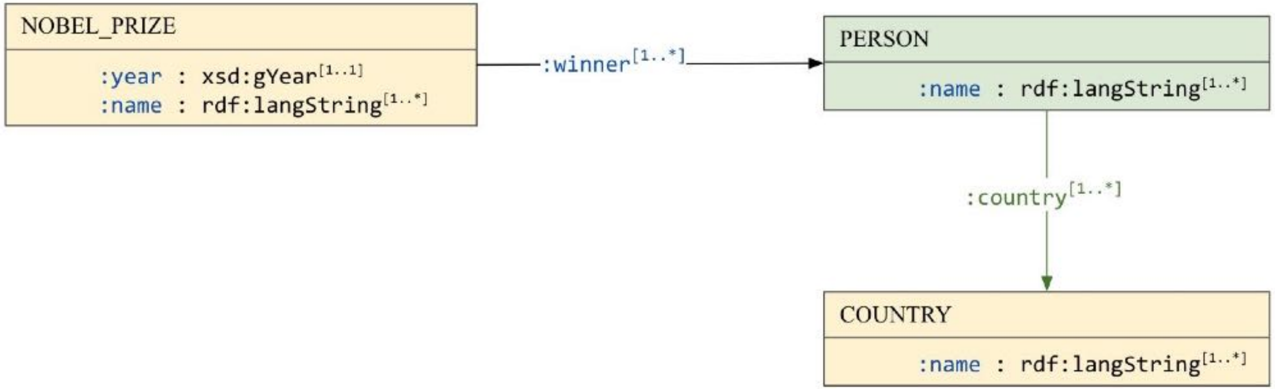
SHACL – Referencing Node Shapes




```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix s: <http://example.org/shapes/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .
@prefix : <http://example.org/> .

s:Nobel_Prize a sh:NodeShape ;
  sh:targetClass :Nobel_Prize ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :year ; sh:datatype xsd:gYear ; sh:minCount 1 ; sh:maxCount 1] ;
  sh:property [sh:path :winner ; sh:node s:Person ; sh:minCount 1 ; sh:class :Person] .
```

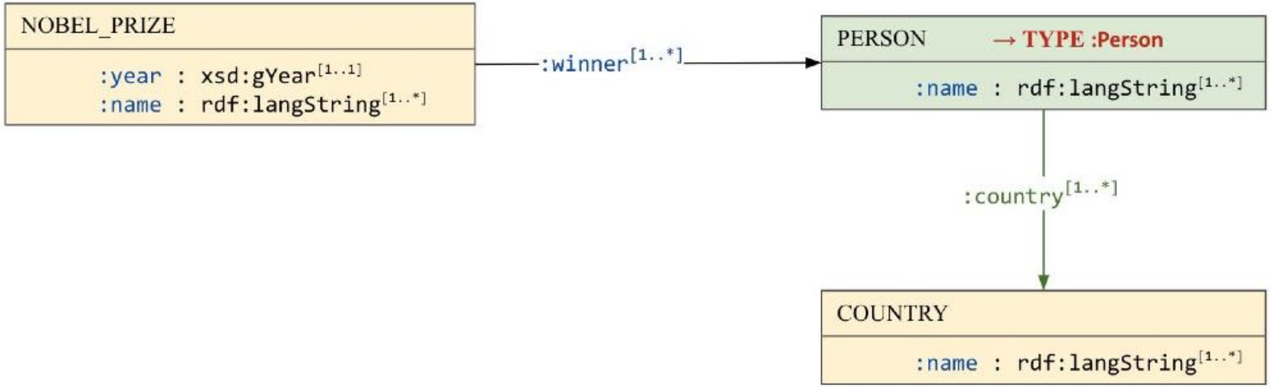
The Shape Constraint Language – SHACL




```
@prefix [...]
s:Nobel_Prize [...]

s:Person a sh:NodeShape ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :country ; sh:node s:Country ; sh:minCount 1 ] .
```

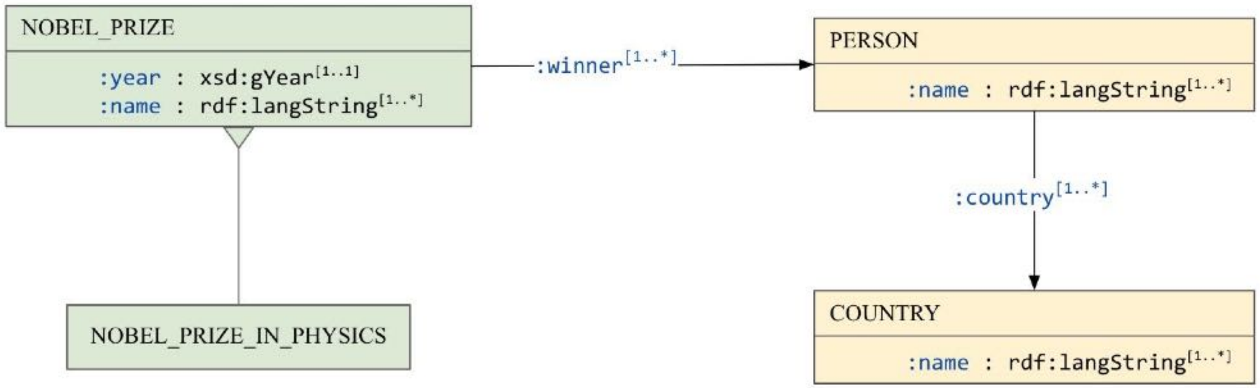
SHACL Targets




```
@prefix [...]
s:Nobel_Prize [...]

s:Person a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :country ; sh:node s:Country ; sh:minCount 1 ] .
```

SHACL Inheritance




```
@prefix [...]
s:Nobel_Prize [...]
s:Person [...]

s:Nobel_Prize_in_Physics a sh:NodeShape ;
  sh:node s:Nobel_Prize .
```

SHACL Complete Example

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix s: <http://example.org/shapes/> .
@prefix : <http://example.org/> .
@prefix sh: <http://www.w3.org/ns/shacl#> .

s:Nobel_Prize a sh:NodeShape ;
  sh:targetClass :Nobel_Prize ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :year ; sh:datatype xsd:gYear ; sh:minCount 1 ; sh:maxCount 1] ;
  sh:property [sh:path :winner ; sh:node s:Person ; sh:minCount 1 ; sh:class :Person] .

s:Nobel_Prize_in_Physics a sh:NodeShape ;
  sh:node s:Nobel_Prize .

s:Person a sh:NodeShape ;
  sh:targetClass :Person ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] ;
  sh:property [sh:path :country ; sh:node s:Country ; sh:minCount 1 ] .

s:Country a sh:NodeShape ;
  sh:targetClass :Country ;
  sh:property [sh:path :name ; sh:datatype rdf:langString ; sh:minCount 1] .
```

SHACL - Hands On

SHACL Play!

https://shacl-play.sparna.fr/play/

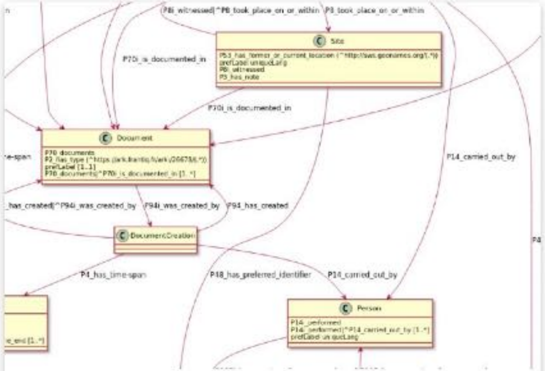
ISE 2022 SHACL GitHub Repository with Example Data

Free online RDF data validation with SHACL. SHACL Play! embeds TopBraid SHACL API from TopQuadrant.

Validate RDF data using SHACL

Get a human-readable report from a SHACL validation. Upload your RDF or validate online RDF file at some URL. Also download a CSV report, or raw SHACL Turtle report.

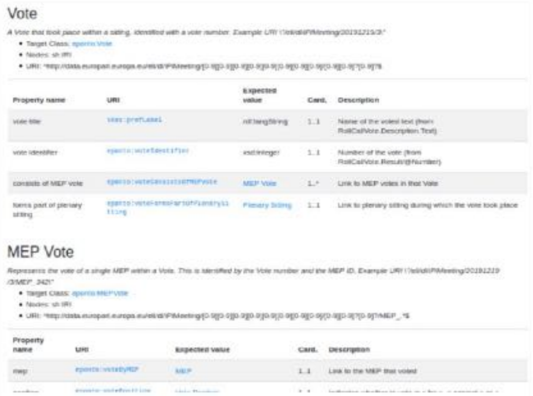
VALIDATE



Draw UML diagrams from Shapes

Looking for something more visual ? Generate UML diagrams in SVG from your SHACL file ! Works with PlantUML

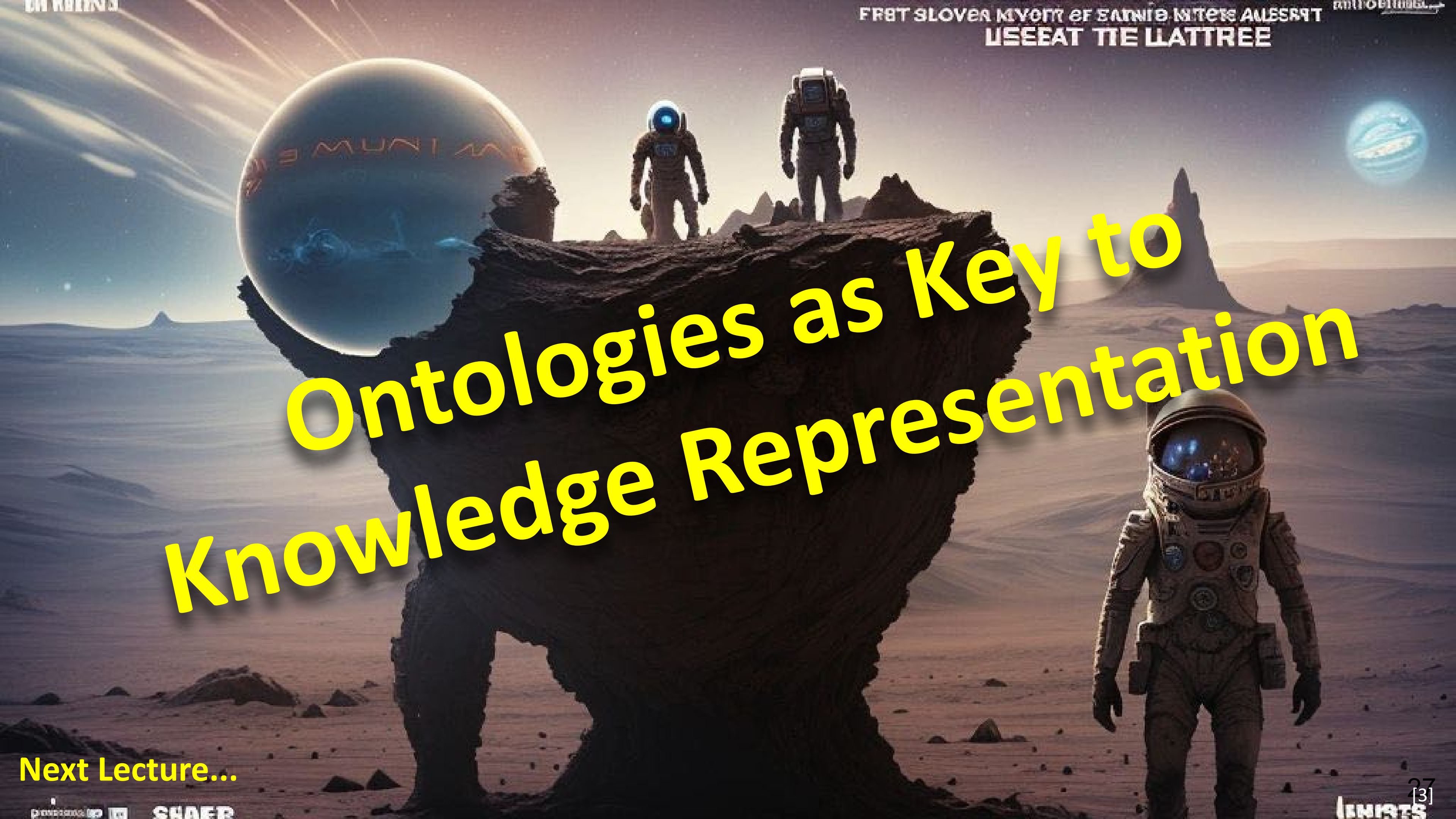
DRAW



Generate documentation

Generate application-profile style documentation, with tables of allowed properties for each class, from your SHACL definition. Ready to be shipped to the developers !

GENERATE DOC



Ontologies as Key to Knowledge Representation

Next Lecture...

[3]

Karlsruher Institut für Technologie
FIZ Karlsruhe
Leibniz Institute for Information Infrastructure

Bibliographic References:

- Holger Knublauch, Dimitris Kontokostas (ed.), *Shapes Constraint Language (SHACL),* W3C Recommendation 20 July 2017
- Jose E. Labra Gayo et al., *Shapes Applications and Tools Tutorial,* ISWC 2020
  SHACL by Example – RDF Validation Tutorial (slides)(video)
- Jose E. Labra Gayo, Eric Prud'hommeaux, Iovka Boneva, Dimitris Kontokostas (2018) Validating RDF Data, Synthesis Lectures on the Semantic Web: Theory and Technology, Vol. 7, No. 1, 1–328, Morgan & Claypool
- Aidan Hogan (2020), *The Web of Data*, Springer.
  Chap. 7.1 Shape Constraint Language - SHACL, pp. 453–500.

Picture References:

[1] "A science fiction movie poster for "Cthulhu and the Gods of Mars" which depicts the first landing of humans on Mars in a retro-futuristic style showing how the great Cthulhu is hovering over the red dessert facing a few human astronauts surrounded by strange ancient artefacts.", created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[2] Benjamin Nowack, *The Semantic Web - Not a Piece of cake…*, at bnode.org, 2009-07-08 , [CC BY 3.0], https://web.archive.org/web/20220628120341/http://bnode.org/blog/2009/07/08/the-semantic-web-not-a-piece-of-cake

[3] Henri Poincaré, unknown author [Public Domain, via WikiCommons, https://commons.wikimedia.org/wiki/File:Henri_Poincar%C3%A9-2.jpg

[4] "A Scifi movie poster of "Planet Mars - the isle of the Dead". A small rover crosses the lonely Martian dessert towards the isle of the dead, on board we see two silent astronauts. Some zombies are chasing after the astronauts.", created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot