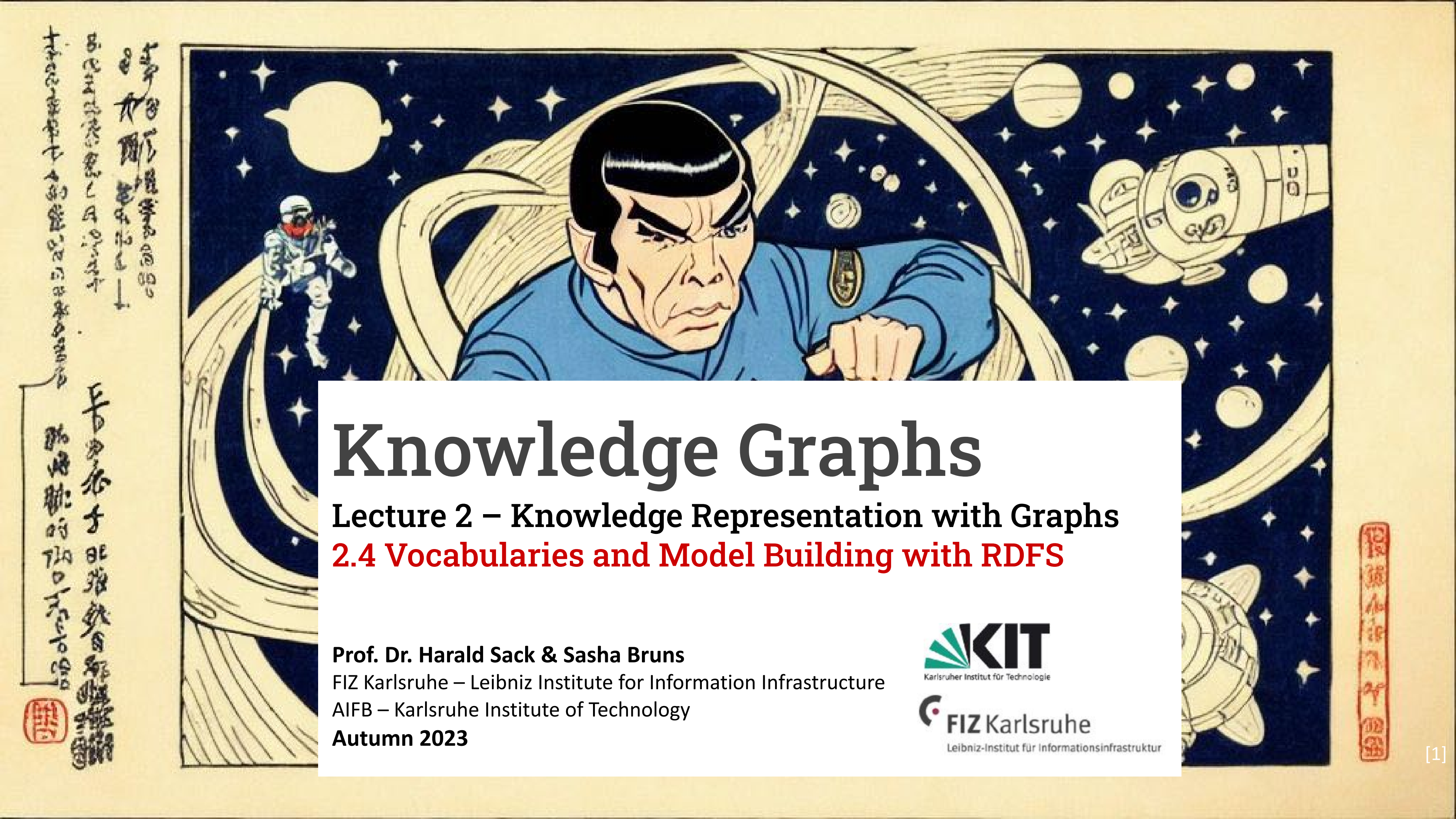


[1]

Knowledge Graphs

Lecture 2 – Knowledge Representation with Graphs

2.4 Vocabularies and Model Building with RDFS

Prof. Dr. Harald Sack & Sasha Bruns
FIZ Karlsruhe – Leibniz Institute for Information Infrastructure
AIFB – Karlsruhe Institute of Technology

Autumn 2023

KIT
Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 2: Basic Knowledge Graph Infrastructure

2.1 How to Identify and Access Things

2.2 How to Represent Simple Facts with RDF

2.3 RDF Turtle Serialization

2.4 Vocabularies and Model Building with RDFS

2.5 RDF Complex Data Structures

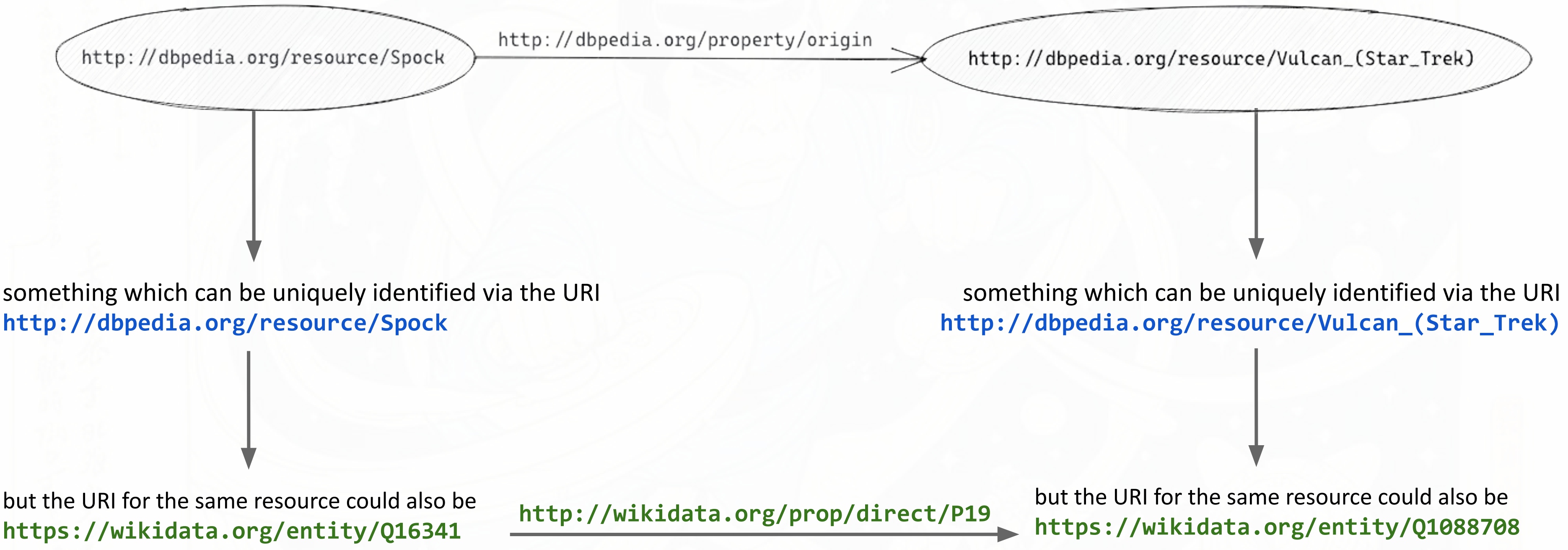
Excursion 1: RDF Reification and RDF*

2.6 Logical Inference with RDF(S)

Excursion 2: RDFa – RDF and the Web

[1]

What does it really mean?



http://dbpedia.org/resource/Spock

http://dbpedia.org/property/origin

http://dbpedia.org/resource/Vulcan_(Star_Trek)

something which can be uniquely identified via the URI
http://dbpedia.org/resource/Spock

something which can be uniquely identified via the URI
http://dbpedia.org/resource/Vulcan_(Star_Trek)

but the URI for the same resource could also be
https://wikidata.org/entity/Q16341

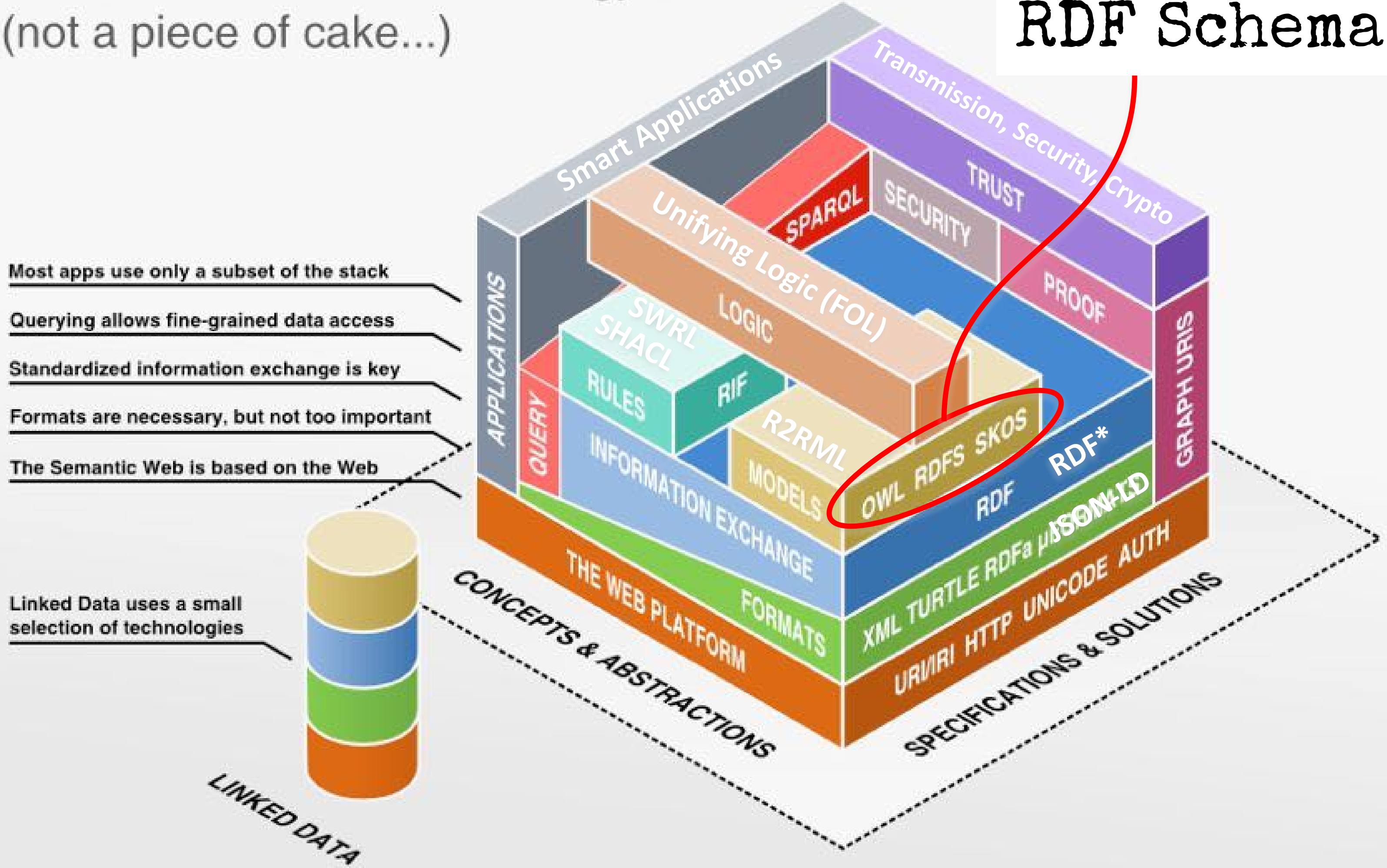
http://wikidata.org/prop/direct/P19

but the URI for the same resource could also be
https://wikidata.org/entity/Q1088708

We need more semantic expressivity…

[1]

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



Most apps use only a subset of the stack
Querying allows fine-grained data access
Standardized information exchange is key
Formats are necessary, but not too important
The Semantic Web is based on the Web
Linked Data uses a small selection of technologies
LINKED DATA
Smart Applications
APPLICATIONS
Transmission, Security, Crypto
TRUST
SECURITY
PROOF
SPARQL
QUERY
Unifying Logic (FOL)
LOGIC
SWRL
SHACL
RULES
RIF
R2RML
MODELS
OWL RDFS SKOS
RDF*
RDF
INFORMATION EXCHANGE
JSON-LD
XML TURTLE RDFa μ
FORMATS
GRAPH URIS
URI/IRI HTTP UNICODE AUTH
THE WEB PLATFORM
CONCEPTS & ABSTRACTIONS
SPECIFICATIONS & SOLUTIONS

RDF Schema

[2]

Terms, Vocabularies, and Schemata

Definitions:

- A **term t** is a word, compound word, or multi-word expression that in specific contexts is given specific meanings.
- A **terminology** or **vocabulary** $V=\{t_1,...,t_n\}$ is a set of terms used to describe data in a particular domain or set of domains.
- A **schema** is a formal description of the high-level structure of a dataset that may be used for a variety of purposes, including managing, storing, indexing, querying, validating, and/or reasoning over a dataset.
- A **semantic schema** is a schema that allows for defining the meaning of high-level terms (aka vocabulary or terminology), which facilitates reasoning over graphs using those terms.

Classes, Properties, and Instances

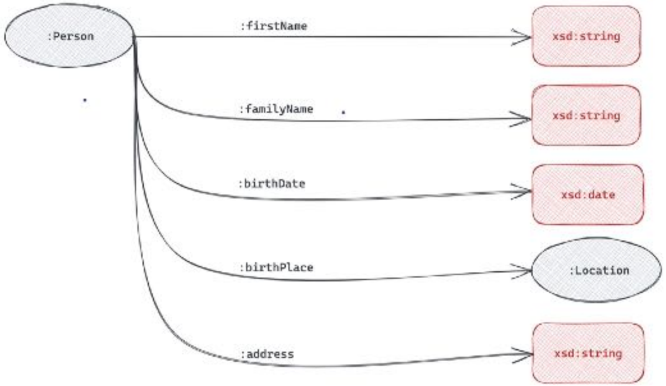
Definitions:

- **Classes** are abstract groups, sets, or collections of resources usually with some conceptional similarities. Classes group resources (**instances**) of the same type.
- **Classes** are characterized by attributes (**properties**).

semi-formal description

Person

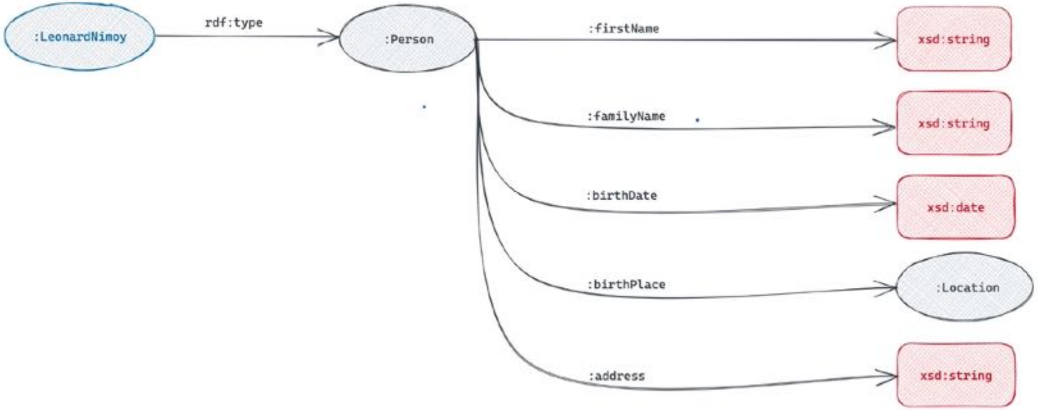
- *firstName <string>*
- *familyName <string>*
- *birthDate <date>*
- *birthPlace <location>*
- *address <string>*
- …



Classes, Properties, and Instances

Definitions:

- An **RDF Property** P is defined as a relation between subject resources S and object resources O, $P \subseteq S \times O$.
- **Instances** are individual members $i_1,\ldots,i_n$ of a class C, $i_1,\ldots,i_n \in C$. An instance can be member of multiple classes.



RDF Schema

- **RDF Schema**, officially called “**RDF Vocabulary Description Language**”

- RDF Schema allows:
  - Definition of **classes** via `rdfs:Class`
  - Class instantiation in RDF via `rdf:type`
  - Example:

```
:Person rdf:type rdfs:Class .
:LeonardNimoy rdf:type :Person .
```

The Class of all classes.

A property to relate an instance to its class.

LeonardNimoy ∈ Person

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix :     <http://example.org/KG2023#> .
```

RDF Schema

- Definition of **properties** via `rdf:Property`

The Class of all properties.

- Definition of **property restrictions on domain and range** via `rdfs:domain` and `rdfs:range`

- Example

```
:Person        rdf:type       rdfs:Class .
:Occupation    rdf:type       rdfs:Class .
:profession    rdf:type       rdf:Property .
:profession    rdfs:domain    :Person .
:profession    rdfs:range     :Occupation .
```

Semantics of `rdfs:domain` and `rdfs:range`:

A,B: Classes, P: Property, a,b: Instances,

$$\forall a,b,A,B,P\text{: } T(a, P, b) \wedge T(P, \texttt{rdfs:domain}, A) \wedge T(P, \texttt{rdfs:range}, B) \rightarrow T(a, \texttt{rdf:type}, A) \wedge T(b, \texttt{rdf:type}, B)$$

RDF Schema

RDFS Meta-Classes:

Everything in the RDF model is a **resource:**

```
○ rdfs:Class        rdf:type rdfs:Resource .
○ rdf:Property      rdf:type rdfs:Resource .
○ rdfs:Literal      rdf:type rdfs:Resource .
○ rdfs:XMLLiteral rdf:type rdfs:Resource .
○ rdfs:Datatype     rdf:type rdfs:Resource .
```

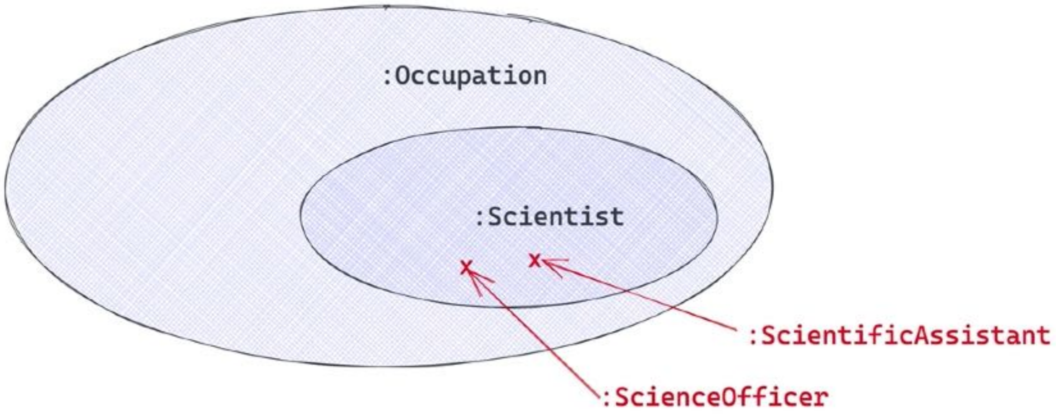
RDF Schema

Definition of **hierarchical relationships**:

- **Subclasses** and **superclasses** via `rdfs:subClassOf`
- Example:

```
:Scientist rdfs:subClassOf :Occupation .
```

Scientist ⊆ occupation



```
:ScientificAssistant rdf:type :Scientist .
:ScienceOfficer rdf:type :Scientist .
```

RDF Schema

Definition of **hierarchical relationships**:

- **Subclasses** and **superclasses** via `rdfs:subClassOf`
- Example:

```
:Scientist rdfs:subClassOf :Occupation .
```

Semantics of `rdfs:subClassOf` (denoted as “$\subseteq$”):

$\forall$A,B: Classes, a: Instance,

$A \subseteq B \leftrightarrow (a \in A \rightarrow a \in B)$

i.e., $\forall a,A,B$: $T(a, \texttt{rdf:type}, A) \land T(A, \texttt{rdfs:subClassOf}, B) \rightarrow T(a, \texttt{rdf:type}, B)$

RDF Schema

Definition of **hierarchical relationships**:

- **Subproperties** and **superproperties** via `subPropertyOf`
- Example:

```
:firstName rdfs:subPropertyOf :name .
```

subproperty

superproperty

Semantics of **rdfs:subPropertyOf** (denoted as “$\subseteq$”):

A,B: Properties, a,b: Instances,

$A \subseteq B \leftrightarrow ((a,b) \in A \rightarrow (a,b) \in B)$

i.e., $\forall$a,b,A,B: T(a, A, b) $\wedge$ T(A, **rdfs:subPropertyOf**, B)$\rightarrow$ T(a, B, b)

RDF Schema

RDFS Annotation Properties

to annotate resources with useful (human-readable) information.

- **`rdfs:seeAlso`**
  defines a relation of a resource to another, which explains it
- **`rdfs:isDefinedBy`**
  subproperty of `rdfs:seeAlso`, defines the relation of a resource to its definition
- **`rdfs:comment`**
  comment, usually as text
- **`rdfs:label`**
  “readable” name of a resource (contrary to ID)

Semantics of `rdfs:seeAlso`, `rdfs:isDefinedBy`, `rdfs:comment`, `rdfs:label` is only denoted via natural language definitions.

RDF Schema

W3C RDFS

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix owl:  <http://www.w3.org/2002/07/owl#> .
@prefix :     <http://example.org/KG2023#> .
```

T-BOX

```
:Person               rdf:type         rdfs:Class .
:FictionalCharacter   rdf:type         rdfs:Class .
:Occupation           rdf:type         rdfs:Class .
:Artist               rdfs:subClassOf  :Occupation .
:Scientist            rdfs:subClassOf  :Occupation .
```

Class Definitions

```
:profession           rdf:type         rdf:Property ;
                      rdfs:domain      :Person, FictionalCharacter ;
                      rdfs:range       :Occupation .
```

Property Definitions

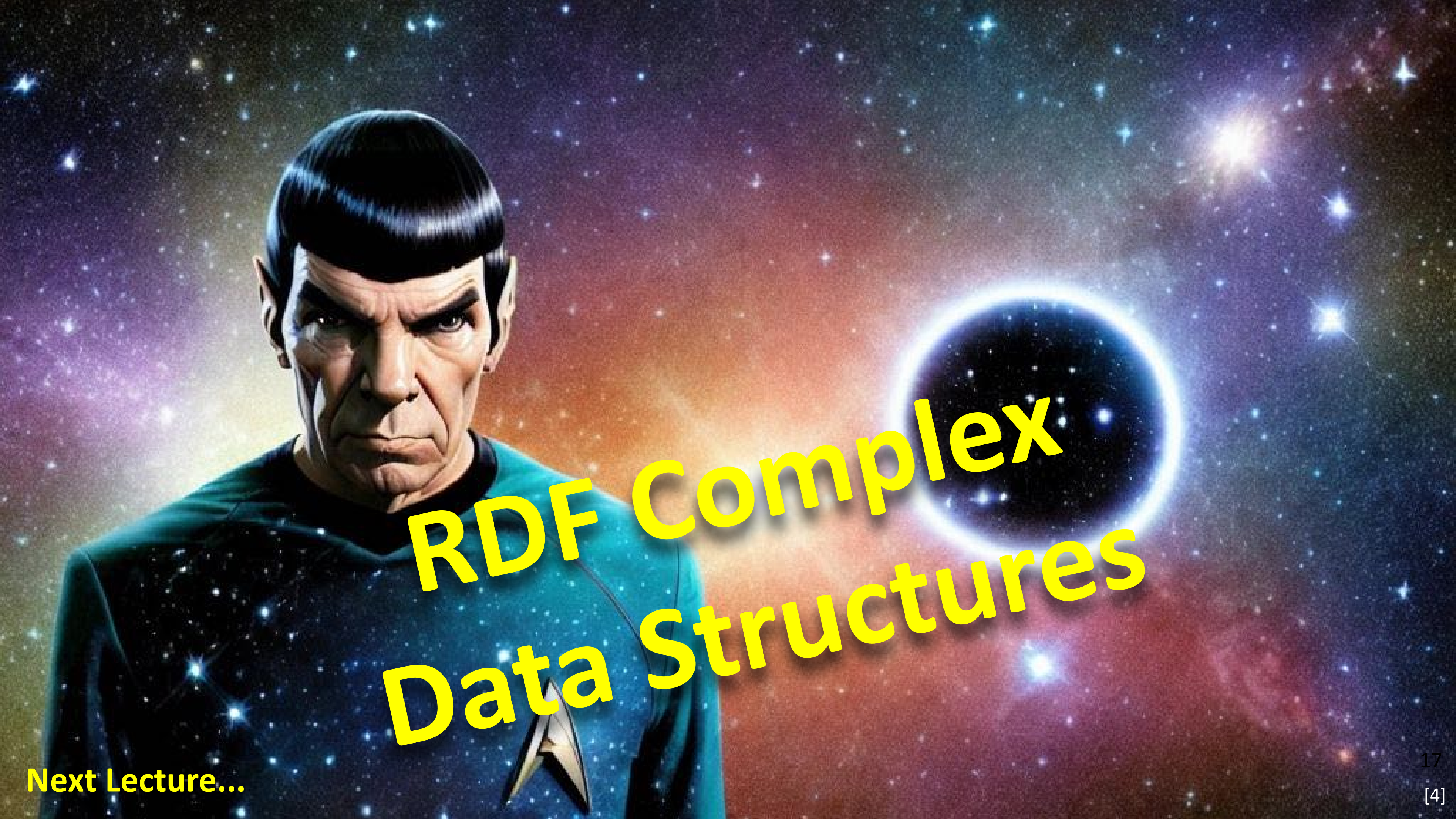
A-BOX

```
:Actor                rdf:type         :Artist .
:ScienceOfficer       rdf:type         :Scientist .
:LeonardNimoy         rdf:type         :Person ;
                      :profession      :Actor ;
                      rdfs:label       Leonard Nimoy”@en .
:Spock                rdf:type         :FictionalCharacter ;
                      rdfs:label       "Spock"@en .
                      profession       :ScienceOfficer .
```

Instance Definitions

RDF Schema - Essentials

- **RDF Classes** and **properties** provide a high-level **vocabulary** – a set of RDF terms – for general use in **RDF descriptions**.
- Vocabularies can be easily reused across different independent RDF sources.
- Datasets that agree on vocabularies are better integrable since they “*speak the same language*”.
- Naming convention:
  - Classes are given **upper case** singular names
    (:Occupation, :Person, :FictionalCharacter, …)
  - Properties are given **lower case** singular names
    (:profession, :birthPlace, :birthDate, …)



Next Lecture...

RDF Complex Data Structures

[4]

Bibliographic References:

- Dan Brickley, R.V. Guha (2014), *RDF Schema 1.1*, W3C Recommendation 25 February 2014.
- Aidan Hogan (2020), *The Web of Data*, Springer.
  Chapt. 3.5 Vocabulary and Modelling, pp. 77–84.

Picture References:

[1] "Mr. Spock, science officer of the USS Enterprise, is fighting a space monster covered with interlinked the RDF source code fragments in the style of a Hokusai woodcut.", created via ArtBot, ProtoGen Diffusion, 2023, [CC-BY-4.0], https://tinybots.net/artbot

[2] LOD Cloud, 2014-08-30, [cc-by-4.0], https://lod-cloud.net/versions/2014-08-30/lod-cloud.png

[3] 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

[4] "In this Star Trek-inspired image, Mr. Spock is depicted in deep space. Among the stars in the background deep space is fully covered with interlinked the RDF code fragments." , created via ArtBot, ProtoGen Diffusion, 2023, [CC-BY-4.0], https://tinybots.net/artbot