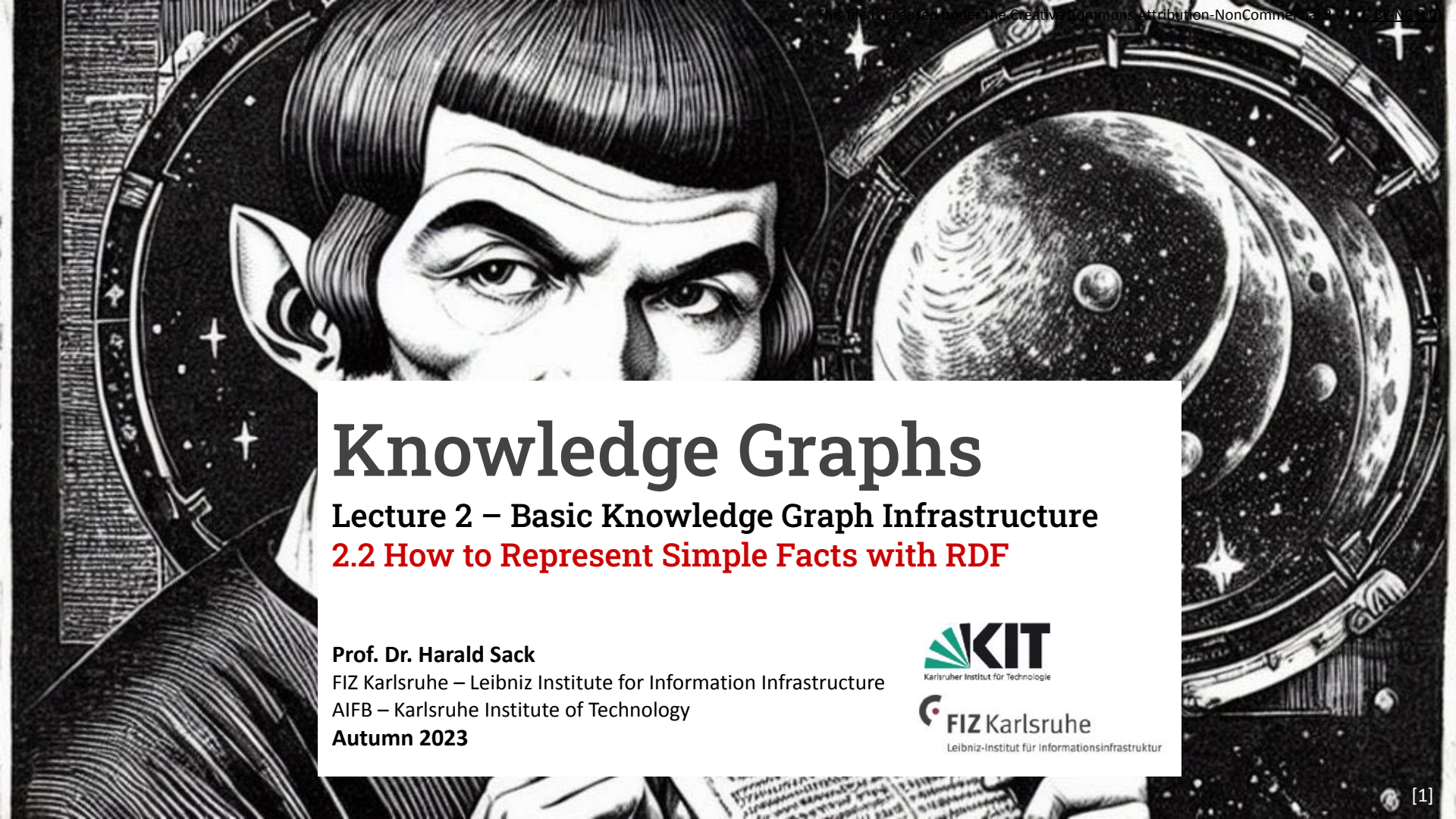
This work is licensed under the Creative Commons Attribution-NonCommercial 4.0 International (CC BY-NC 4.0)



Knowledge Graphs

Lecture 2 – Basic Knowledge Graph Infrastructure

2.2 How to Represent Simple Facts with RDF

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

Karlsruher Institut für Technologie

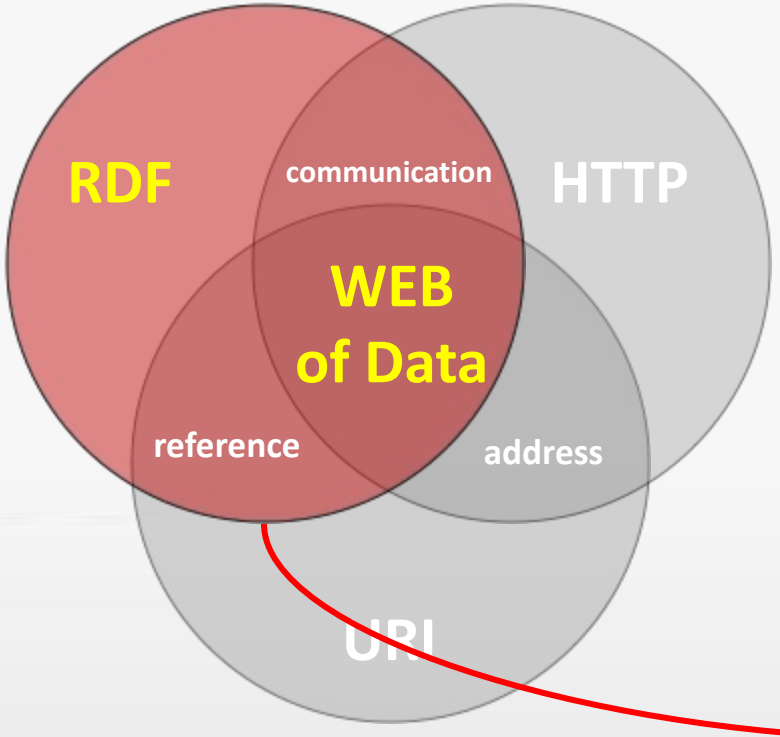
FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

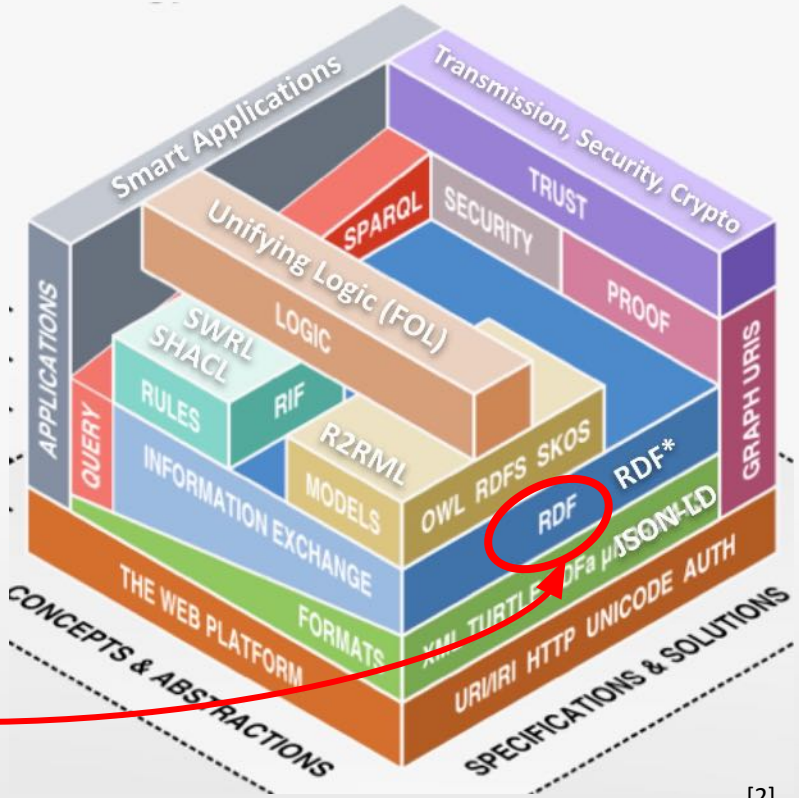
[1]

Knowledge Graphs

Lecture 2: Basic Knowledge Graph Infrastructure

Basic Architecture of the Web of Data

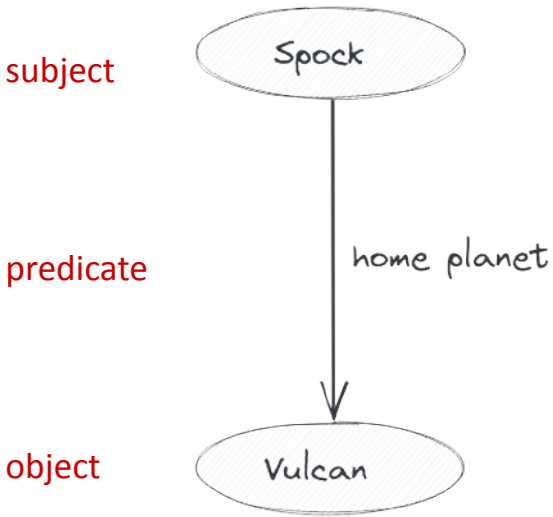




[2]

Intuitive Knowledge Representation with Graphs

How do I represent the following fact:
"***Spock's home planet is Vulcan***" in an intuitive way?



Resource Description Framework

How do I represent the following fact:
"***Spock's home planet is Vulcan***" in an intuitive way?

Resource Description Framework

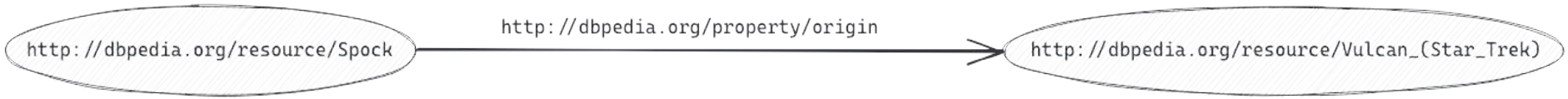
In RDF the predicate of a statement is referred to as "Property".

RDF Statements (RDF-Triple):

Subject	Property	Object / Value
URI	**URI**	**URI / Literal**

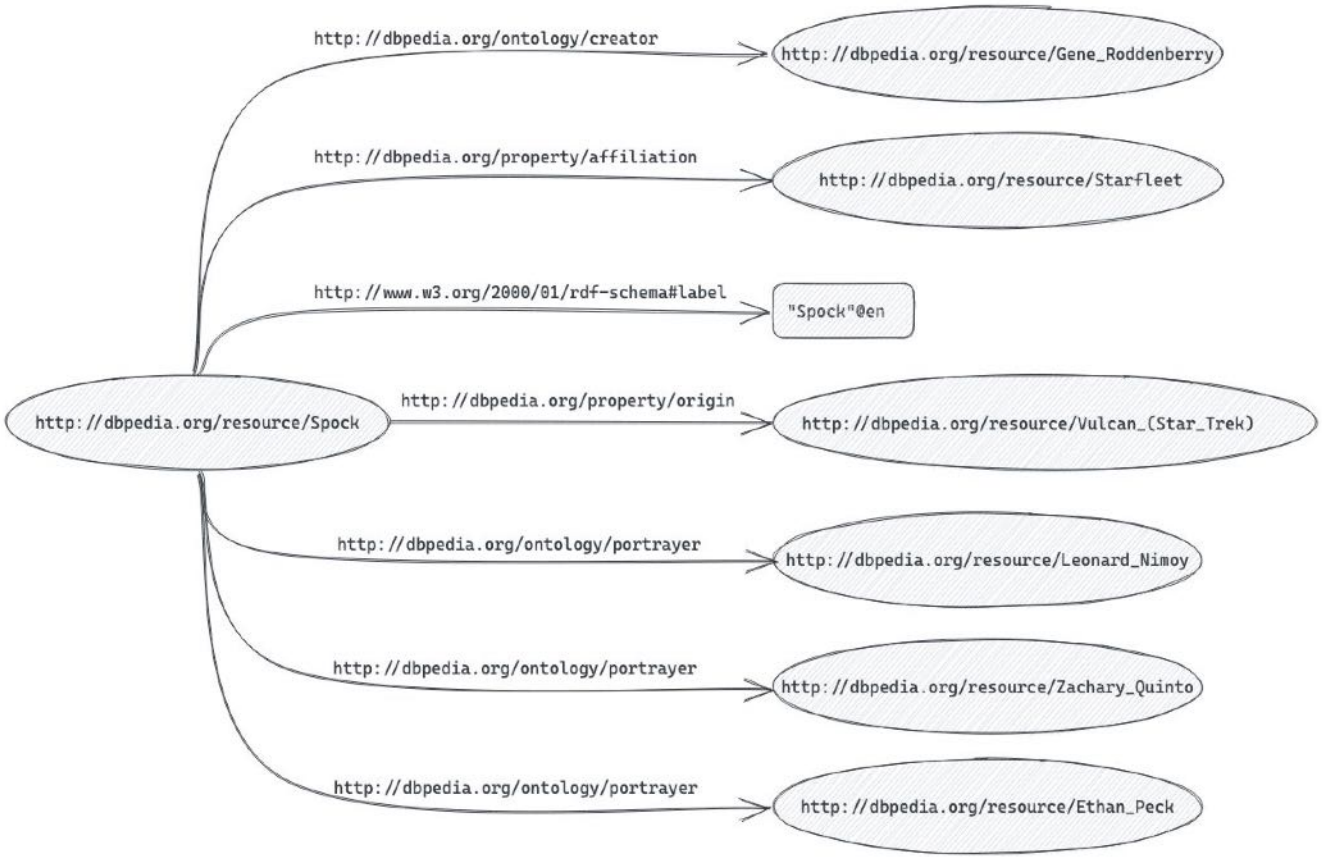
N-Triples Serialization

```
<http://dbpedia.org/resource/Spock>  <http://dbpedia.org/property/origin>  <http://dbpedia.org/resource/Vulcan_(Star_Trek)>
```



Graph Representation

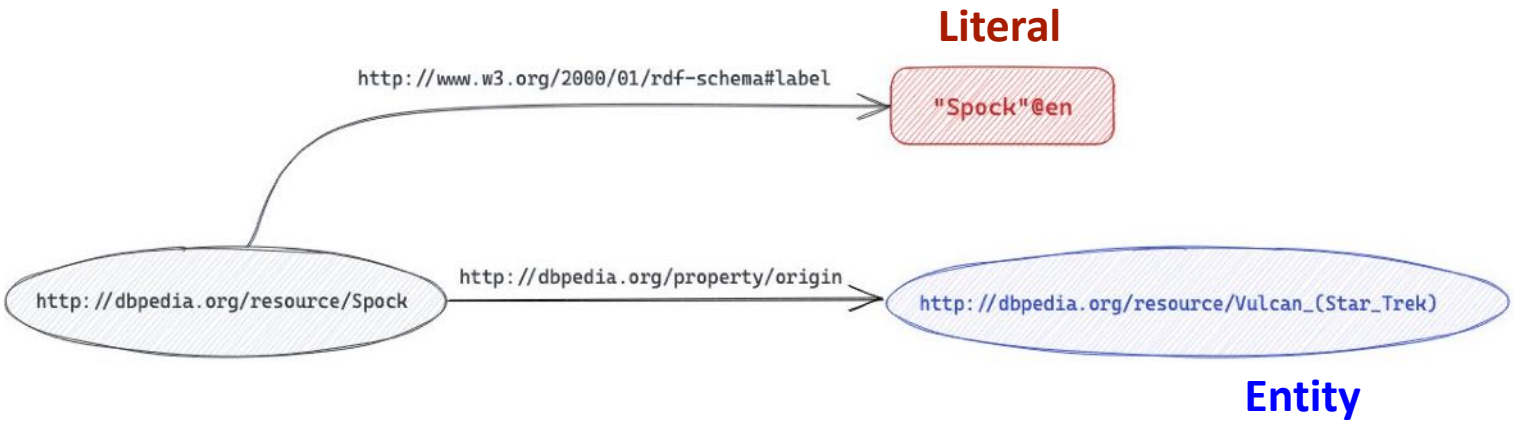
Resource Description Framework



Resource Description Framework

URIs and Literals

- **URIs** identify and reference resources uniquely.
- **Literals** describe data values that don't have a separate existence.



RDF Literals and Datatypes

- Typed literals can be expressed via **XML Schema datatypes.**
- Namespace for typed literals:
  `http://www.w3.org/2001/XMLSchema#`

- Examples:

```
"Spock"^^<http://www.w3.org/2001/XMLSchema#string>
"1161.00"^^<http://www.w3.org/2001/XMLSchema#float>
"2023-08-02"^^<http://www.w3.org/2001/XMLSchema#date>
```

- **Language Tags** denote the (natural) language of the text:

  Example:

```
"Semantik"@de , "Semantics"@en
```

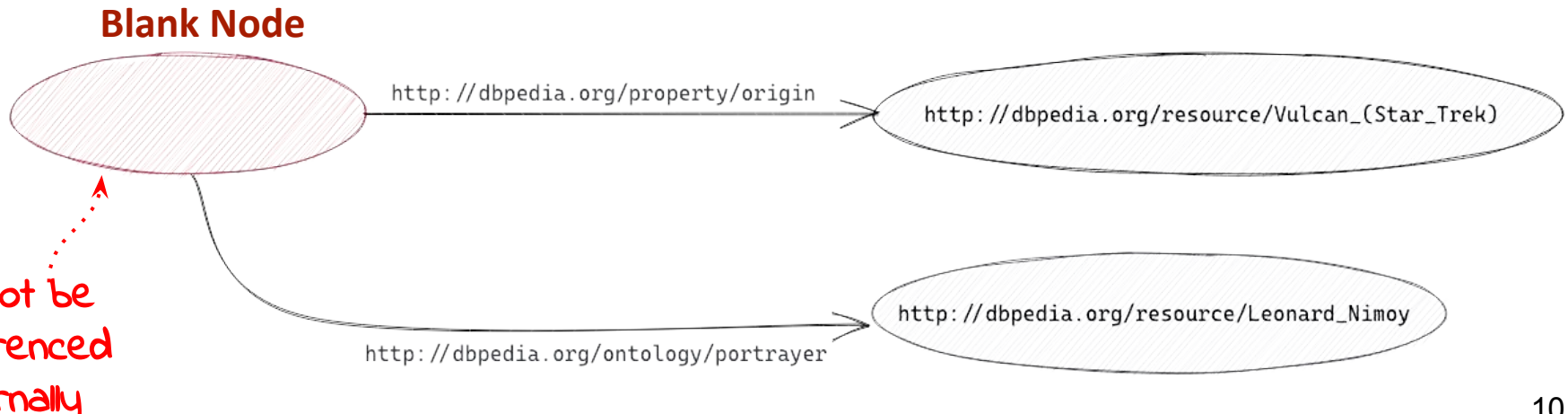
https://www.w3.org/TR/rdf11-concepts/#section-Datatypes

Core types	xsd:string	Character strings
	xsd:boolean	true, false
	xsd:decimal	Arbitrary-precision decimal
	xsd:integer	Arbitrary-size integer numbe
IEEE floating-point numbers	xsd:double	64-bit floating point numbers
	xsd:float	32-bit floating point numbers
Time and date	xsd:date	Dates (yyyy-mm-dd) with or
	xsd:time	Times (hh:mm:ss.sss…) with
	xsd:dateTime	Date and time with or withou
	xsd:dateTimeStamp	Date and time with required
Recurring and partial dates	xsd:gYear	Gregorian calendar year
	xsd:gMonth	Gregorian calendar month
	xsd:gDay	Gregorian calendar day of th
	xsd:gYearMonth	Gregorian calendar year and
	xsd:gMonthDay	Gregorian calendar month a
	xsd:duration	Duration of time
	xsd:yearMonthDuration	Duration of time (months and
	xsd:dayTimeDuration	Duration of time (days, hours
Limited-range integer numbers	xsd:byte	-128…+127 (8 bit)
	xsd:short	-32768…+32767 (16 bit)
	xsd:int	-2147483648…+2147483647
	xsd:long	-9223372036854775808…
	xsd:unsignedByte	0…255 (8 bit)
	xsd:unsignedShort	0…65535 (16 bit)
	xsd:unsignedInt	0…4294967295 (32 bit)
	xsd:unsignedLong	0…18446744073709551615
	xsd:positiveInteger	Integer numbers >0
	xsd:nonNegativeInteger	Integer numbers ≥0
	xsd:negativeInteger	Integer numbers <0
	xsd:nonPositiveInteger	Integer numbers ≤0
Encoded binary data	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary dat
Miscellaneous XSD types	xsd:anyURI	Absolute or relative URIs an
	xsd:language	Language tags per [BCP47]
	xsd:normalizedString	Whitespace-normalized strin
	xsd:token	Tokenized strings
	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
	xsd:NCName	XML NCNames

Existential Assertions – Blank Nodes

Blank Nodes

denote the **existence of an individual** with specific attributes, but **without providing an identification or reference.**



RDF Terms, RDF Triples, and RDF Graph

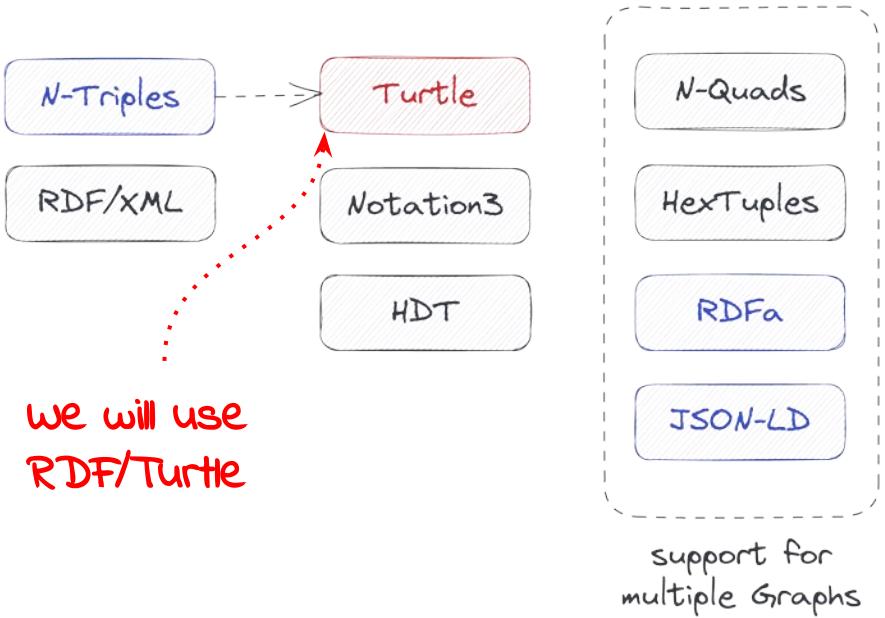
Definitions:

- Let **I** denote the set of IRIs, **L** the set of RDF Literals and **B** the set of RDF blank nodes. The set of **RDF terms** is defined as $\mathbf{I \cup L \cup B}$.
- An **RDF triple** **t:=(s,p,o)** is any element of the set $(I \cup B) \times (I) \times (I \cup B \cup L)$, where $s \in (I \cup B)$ is called the subject,
  $p \in I$ is called the predicate and
  $o \in (I \cup B \cup L)$ is called the object.
- An **RDF graph** G is a subset of $(I \cup B) \times (I) \times (I \cup B \cup L)$,
  i.e. an RDF graph is a set of RDF triples.

RDF Serializations

RDF comes with several different **serialization formats**:

N-Triples, RDF/XML, JSON-LD, Turtle, N-Quads, RDFa, Notation3, HexTuples...



Next Lecture...

[4]

Bibliographic References:

- Guus Schreiber, Yves Raimond (2014), *RDF 1.1 Primer*, W3C Working Group Note 24 June 2014
- Joem Madertma (2019), *What's the best RDF serialization format?*, ontola.io
- *RDFSyntax,* W3CWiki

Picture References:

[1] "The Resource Description Framework (RDF) is a W3C standard originally designed as a data model for metadata…" , created via ArtBot, Anything Diffusion - generic scifi, 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 comic book-style illustration, the Teenage Mutant Ninja Turtles are depicted meeting Mr. Spock, the science officer of the USS Enterprise. The image shows information about each turtle, including their names, abilities, and relationships to one another, using RDF's standardized method for describing and linking resources." , created via ArtBot, Protogen Diffusion, 2023, [CC-BY-4.0], https://tinybots.net/artbot