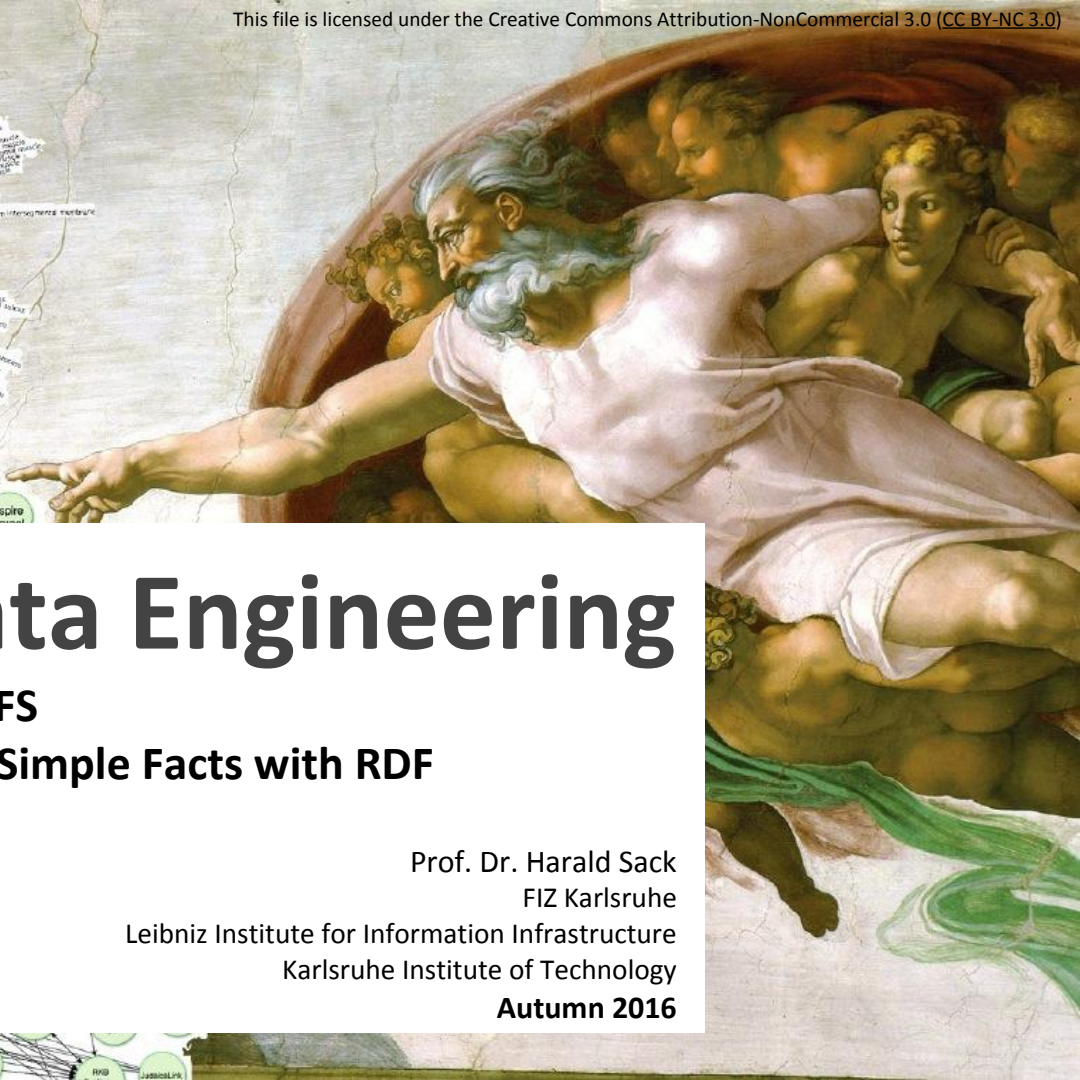




Lecture 2: RDF and RDFS

2.1 How to Represent Simple Facts with RDF



Leibniz Institute for Information Infrastructure

Prof. Dr. Harald Sack
FIZ Karlsruhe

Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Autumn 2016

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

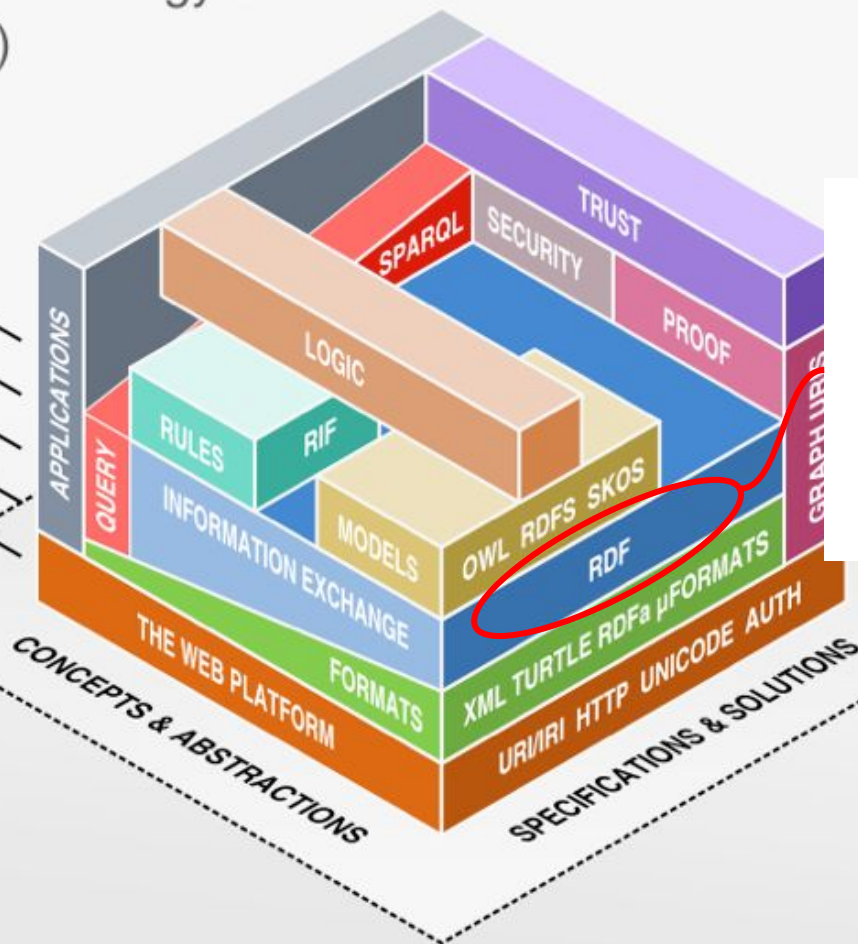
Standardized information exchange is key

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



Resource
Description
Framework
(RDF)

Simple Facts with RDF



PLUTO
Never Forget!
(1930–2006)



Resource Description Framework

- **Resource**
 - can be everything
 - must be uniquely **identified** and **referenceable** via **URI**
- **Description**
 - = description of resources
 - via representing properties and relationships among resources as **graphs**
- **Framework**
 - = combination of web based protocols (URI, HTTP, XML, Turtle, JSON, ...)
 - based on formal model (semantics)
- Knowledge in RDF is expressed as a **list of statements**
- all RDF statements follow the same simple schema (= **RDF Triple**)



Resource Description Framework

- **RDF Statements (RDF Triple):**

Subject + Property + Object / Value

URI

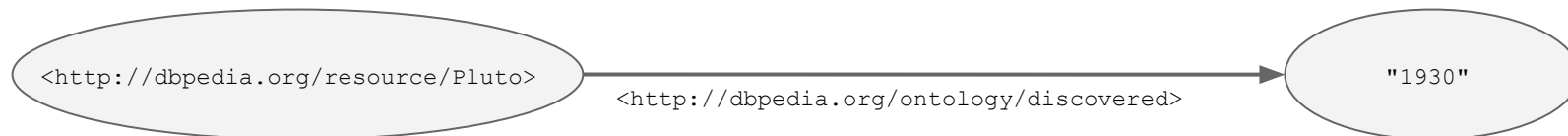
URI

URI / Literal

RDF Building Blocks

N-Triples Serialization

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .
```



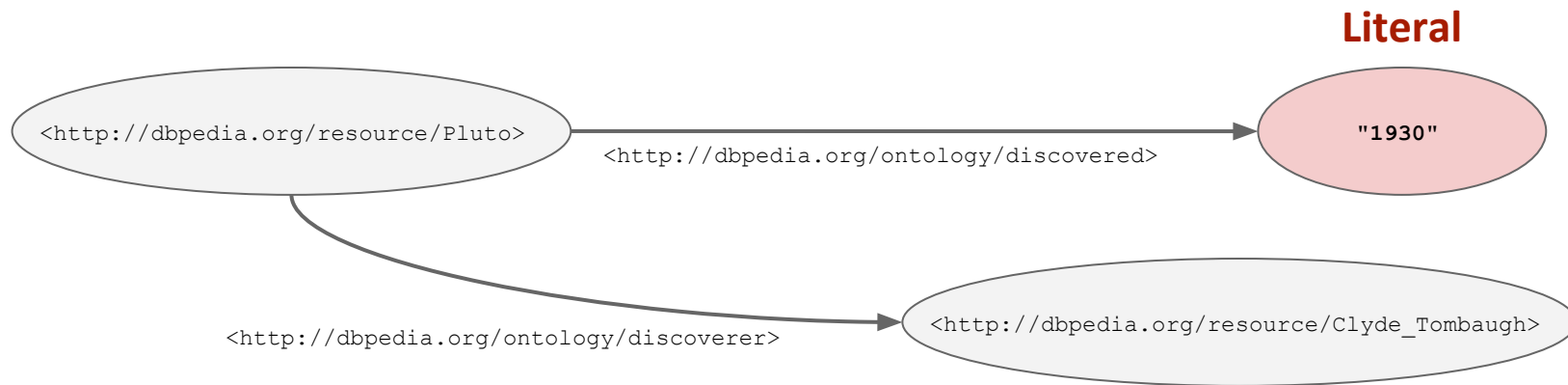
graph
representation



Resource Description Framework

- **URIs and Literals**

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





Literals and Datatypes

- Typed literals can be expressed via **XML Schema datatypes**
- Namespace for typed literals:

<http://www.w3.org/2001/XMLSchema#>

- Examples:

"Semantics"^^<<http://www.w3.org/2001/XMLSchema#string>>

"1161.00"^^<<http://www.w3.org/2001/XMLSchema#float>>

"2015-08-02"^^<<http://www.w3.org/2001/XMLSchema#date>>

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

- Example:

"Semantik"@de , "Semantics"@en

<http://www.w3.org/TR/2013/WD-rdf11-concepts-20130115/#xsd-datatypes>

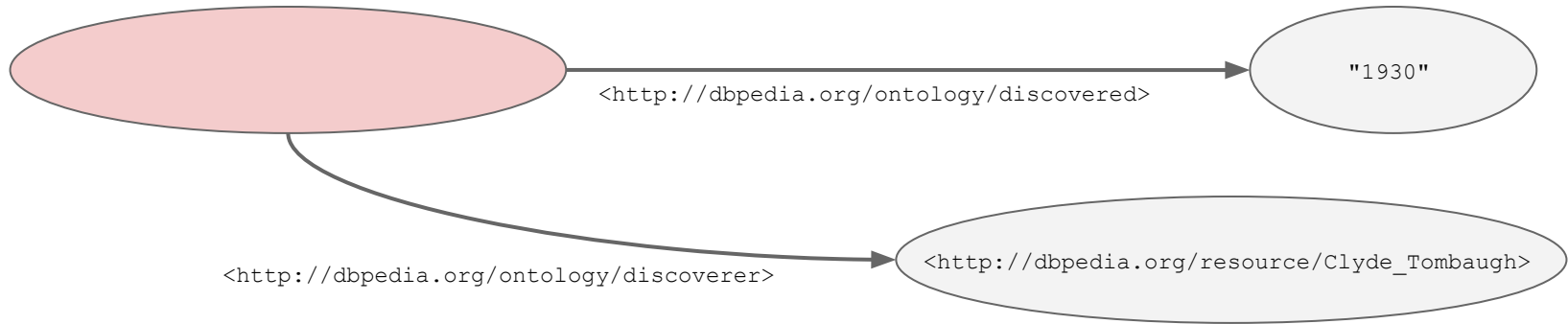
Core types	xsd:string	Character strings
	xsd:boolean	true, false
	xsd:decimal	Arbitrary-precision decimal
	xsd:integer	Arbitrary-size integer numbers
IEEE floating-point numbers	xsd:double	64-bit floating point numbers
	xsd:float	32-bit floating point numbers
Time and date	xsd:date	Dates (yyyy-mm-dd) with or without time
	xsd:time	Times (hh:mm:ss.sss...) with or without time zone
	xsd:dateTime	Date and time with or without time zone
	xsd:dateTimeStamp	Date and time with required time zone
Recurring and partial dates	xsd:gYear	Gregorian calendar year
	xsd:gMonth	Gregorian calendar month
	xsd:gDay	Gregorian calendar day of the month
	xsd:gYearMonth	Gregorian calendar year and month
	xsd:gMonthDay	Gregorian calendar month and day
	xsd:duration	Duration of time
	xsd:yearMonthDuration	Duration of time (months and years)
	xsd:dayTimeDuration	Duration of time (days, hours, minutes, seconds)
Limited-range integer numbers	xsd:byte	-128...+127 (8 bit)
	xsd:short	-32768...+32767 (16 bit)
	xsd:int	-2147483648...+2147483647
	xsd:long	-9223372036854775808...+9223372036854775807
	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
Encoded binary data	xsd:negativeInteger	Integer numbers <0
	xsd:nonPositiveInteger	Integer numbers ≤0
	xsd:hexBinary	Hex-encoded binary data
	xsd:base64Binary	Base64-encoded binary data
Miscellaneous XSD types	xsd:anyURI	Absolute or relative URIs and IRI references
	xsd:language	Language tags per [BCP47]
	xsd:normalizedString	Whitespace-normalized strings
	xsd:token	Tokenized strings
	xsd:NMTOKEN	XML NMTOKENs
	xsd:Name	XML Names
	xsd:NCName	XML NCNames



Blank Nodes

- **Blank Nodes**
 - denote **existence of an individual** with specific attributes, but **without providing an identification or reference**

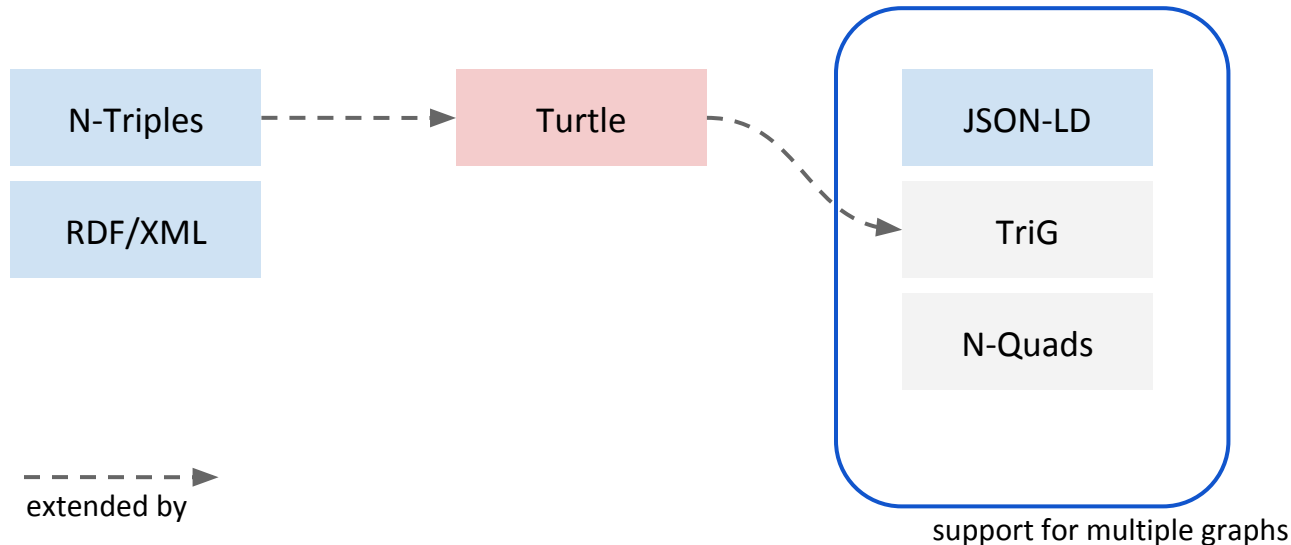
Blank Node





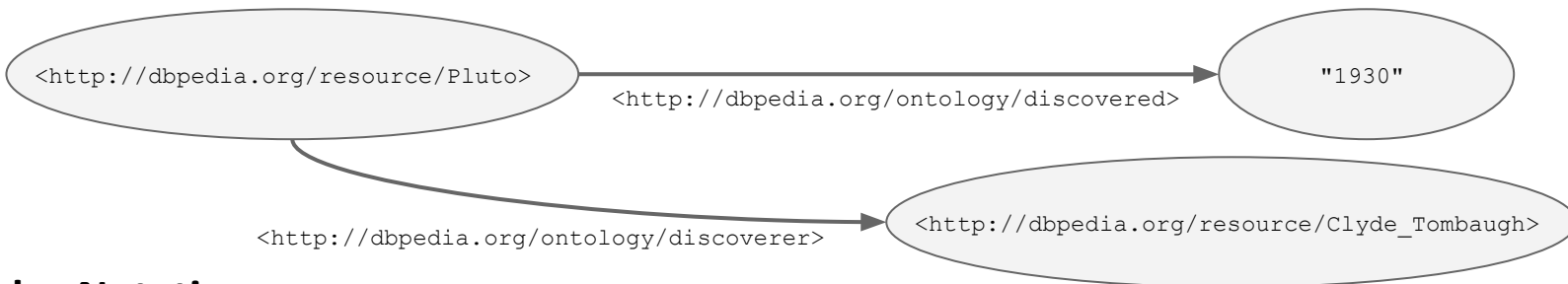
RDF Serializations

- RDF comes with several different **serialization formats**:
 - N-Triples, RDF/XML, JSON, Turtle, TriG, N-Quads, RDFa, ...





RDF Serializations



- **N-Triples Notation**

- **URIs/IRIs** in angle brackets
- **Literals** in quotation marks
- Triple ends with a **period**

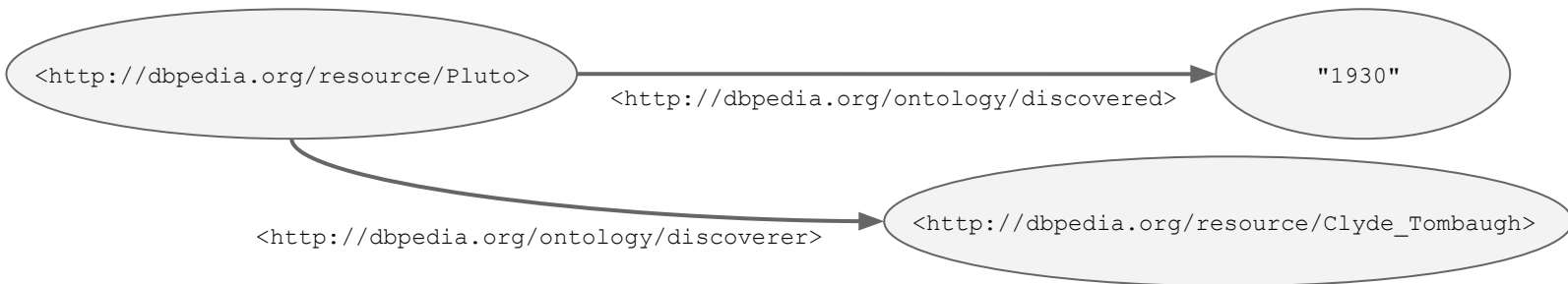
```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discovered> "1930" .
```

```
<http://dbpedia.org/resource/Pluto> <http://dbpedia.org/ontology/discoverer>
```

```
<http://dbpedia.org/resource/Clyde_Tombaugh> .
```



RDF Serializations



- **RDF/XML Notation**

S

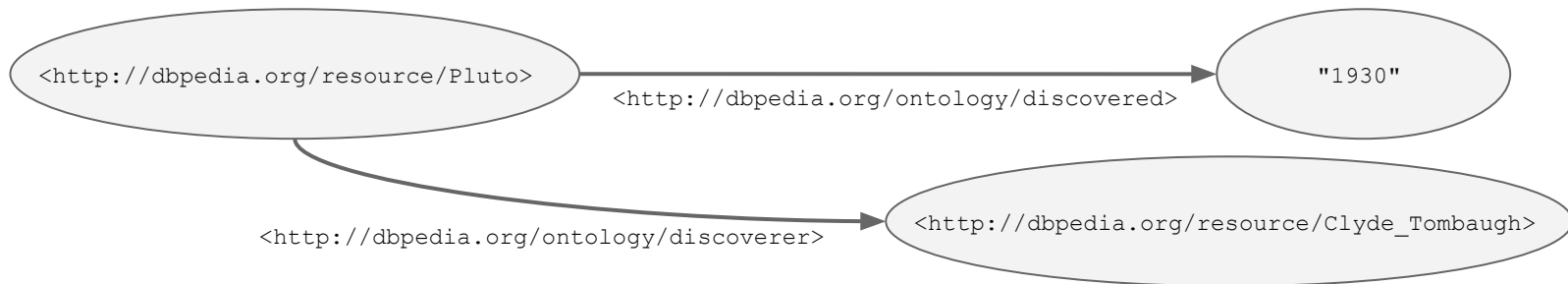
P

O

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ns0="http://dbpedia.org/ontology/">
  <rdf:Description rdf:about="http://dbpedia.org/resource/Pluto">
    <ns0:discovered>1930</ns0:discovered>
    <ns0:discoverer rdf:resource="http://dbpedia.org/resource/Clyde_Tombaugh"/>
  </rdf:Description>
</rdf:RDF>
```



RDF Serializations



- **JSON-LD Notation (RDF 1.1)**

S

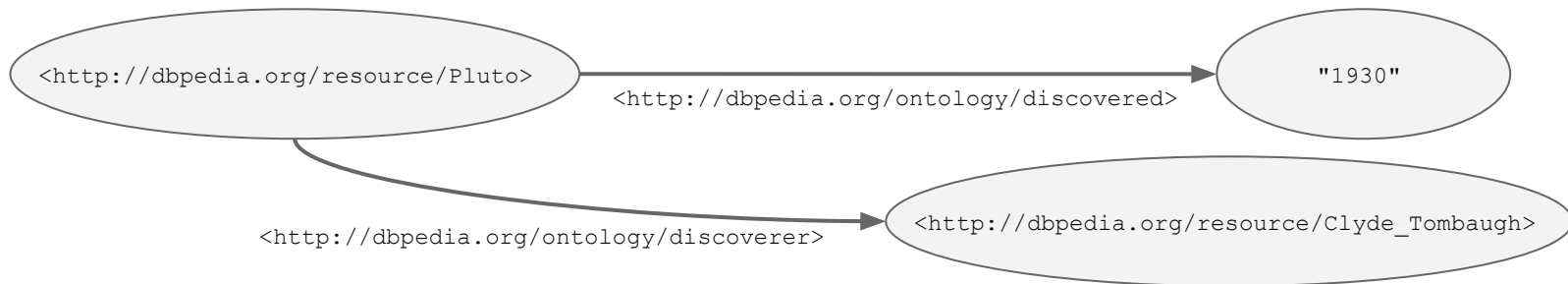
P

O

```
{  "@id" : "http://dbpedia.org/resource/Pluto" ,
  "http://dbpedia.org/ontology/discovered" :
    { "@value" : "1930" }
,
  "http://dbpedia.org/ontology/discoverer" :
    { "@id" : "http://dbpedia.org/resource/Clyde_Tombaugh" }
}
```



RDF Serializations



- **Turtle (Terse RDF Triple Language) Notation**
 - extension of N-Triples

```
@prefix dbo: <http://dbpedia.org/ontology/> .  
@base <http://dbpedia.org/resource/> .  
  
<Pluto> dbo:discovered "1930" .  
  
<Pluto> dbo:discoverer <Clyde_Tombaugh> .
```



Next: 02 - RDF and Turtle Serialization

OpenHPI - Course Linked Data Engineering - Lecture 2: RDF and RDFS