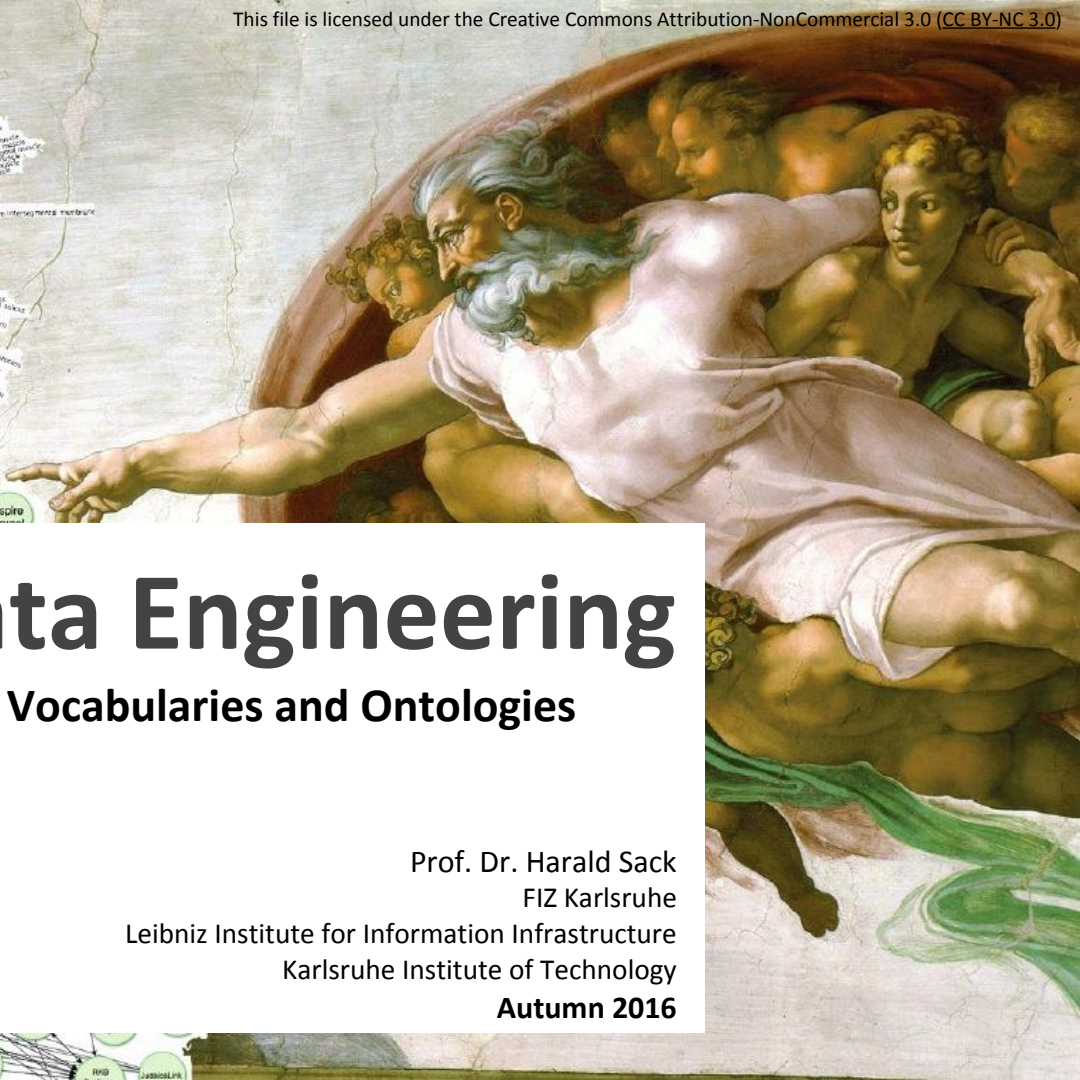




Lecture 3: Linked Data Vocabularies and Ontologies

3.3 First Steps in OWL

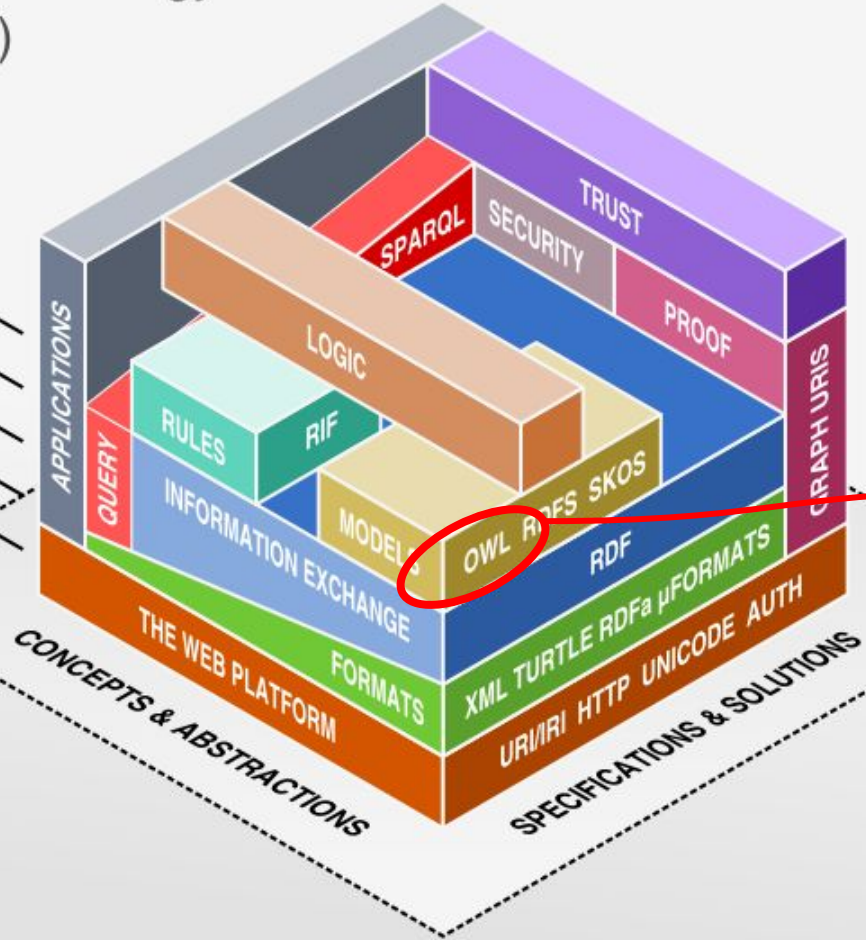
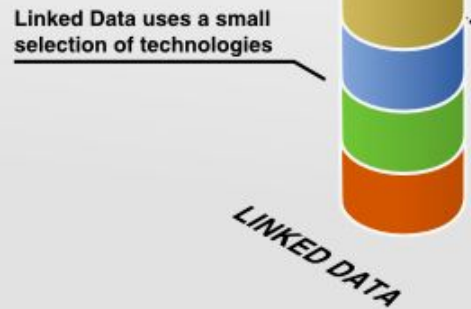


FIZ Karlsruhe

Leibniz Institute for Information Infrastructure
Karlsruhe Institute of Technology
Autumn 2016

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

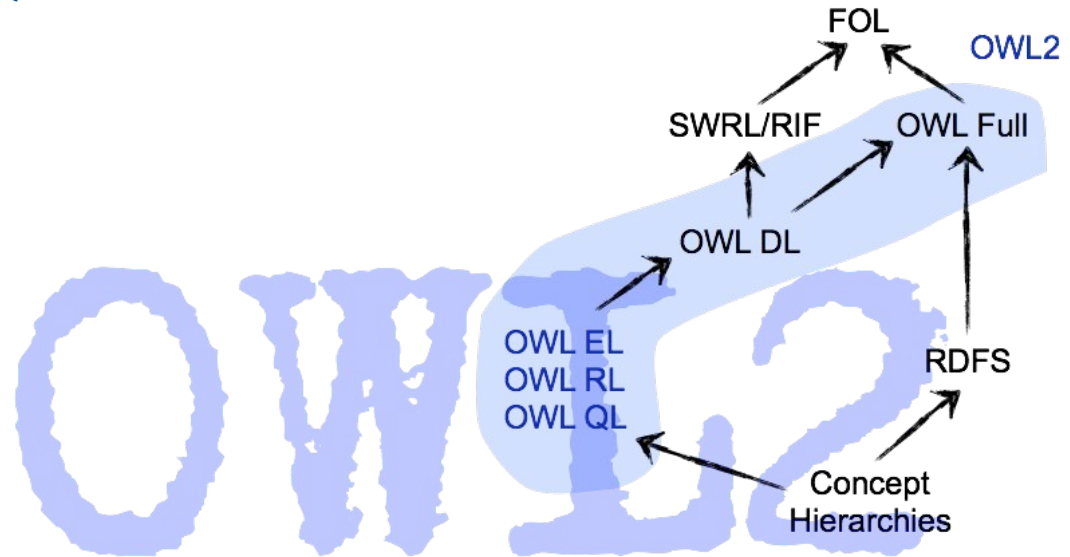
- Most apps use only a subset of the stack
- Querying allows fine-grained data access
- Standardized information exchange is key
- Formats are necessary, but not too important
- The Semantic Web is based on the Web



Web Ontology Language (OWL)

Web Ontology Language - OWL

- OWL is a semantic fragment of First Order Logic (FOL)
- OWL also exists in different flavors
 - $\text{OWL EL, OWL RL, OWL QL} \subseteq \text{OWL2 DL} \subseteq \text{OWL2 Full}$



OWL2 is based on the Description Logic $\mathcal{SROIQ}(\mathcal{D})$

Class Expressions

- Class names A, B
- Conjunction $C \sqcap D$
- Disjunction $C \sqcup D$
- Negation $\neg C$
- Exist. property restriction $\exists R.C$
- Univ. property restriction $\forall R.C$
- Self $\exists S.\text{Self}$
- Greater-than $\geq n \ S.C$
- Less-than $\leq n \ S.C$
- Enumerated classes $\{a\}$

Properties

- Property names R, S, T
- Simple properties S, T
- Inverse properties R^{-}
- Universal property U

Tbox (Class axioms)

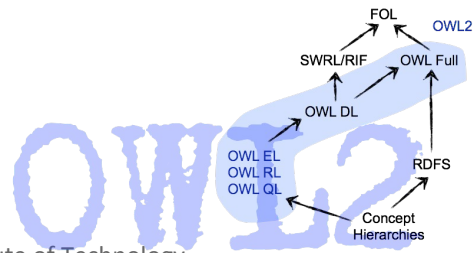
- Inclusion $C \sqsubseteq D$
- Equivalence $C \equiv D$

Rbox (Property Axioms)

- Inclusion $R_1 \sqsubseteq R_2$
- General Inclusion $R^{(-)}_1 \circ R^{(-)}_2 \circ \dots \circ R^{(-)}_n \sqsubseteq R$
- Transitivity
- Symmetry
- Reflexivity
- Irreflexivity
- Disjunctiveness

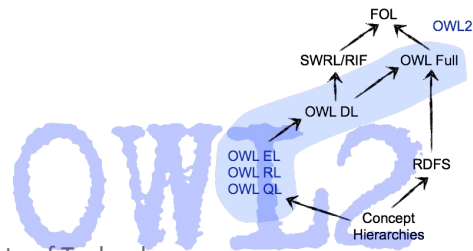
Abox (Facts)

- Class membership $C(a)$
- Property relation $R(a, b)$
- Negated property relation $\neg S(a, b)$
- Equality $a=b$
- Inequality $a \neq b$



OWL Basic Building Blocks

- **OWL namespace:**
`@prefix owl: <http://www.w3.org/2002/07/owl#>`
- There is a **Turtle Syntax** for OWL
- OWL axioms consist of the following three building blocks:
 - **Classes**
 - comparable with classes in RDFS
 - **Individuals**
 - comparable with class instances in RDFS
 - **Properties**
 - comparable with properties in RDFS



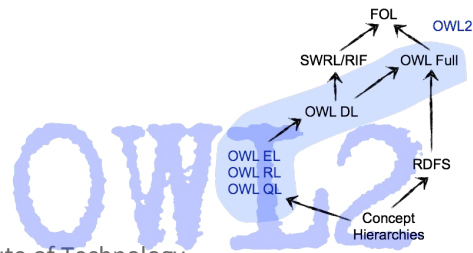
OWL Classes

- there exist two predefined classes
 `owl:Thing` (class that contains all individuals)
 `owl:Nothing` (empty class)
- Definition of a class
 :Book a `owl:Class` .

equivalent expression in
description logics

$$\top \equiv C \sqcup \neg C$$

$$\perp \equiv C \sqcap \neg C$$

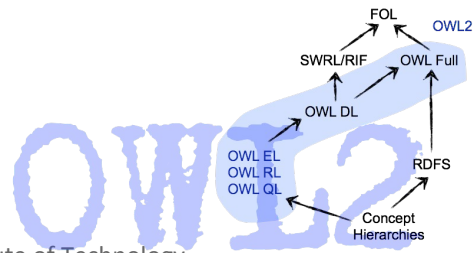


OWL Individuals

- Definition of individuals via class membership
`:NineteenEightyfour a :Book .`

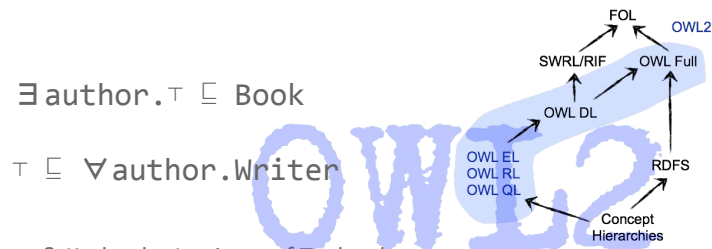
`Book(NineteenEightyfour)`

- Individuals can also be defined without class membership
as named individuals
`:HaraldSack a owl:NamedIndividual .`



OWL Object Properties

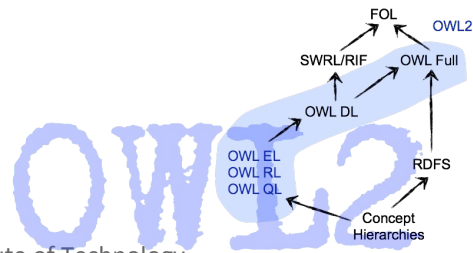
- there exist two property variants:
 - object properties
 - datatype properties
- **Object properties** have classes as range
:author a `owl:ObjectProperty` .
- Domain and Range of object properties
:author a `owl:ObjectProperty` ;
rdfs:domain :Book ;
rdfs:range :Writer .



OWL Datatype Properties

- **Datatype properties** have datatypes as range
:publicationYear a owl:DatatypeProperty .

- Domain and Range of datatype properties
:publicationYear a owl:DatatypeProperty ;
rdfs:domain :Book ; $\exists \text{publicationYear}.\top \sqsubseteq \text{Book}$
rdfs:range xsd:integer . $\top \sqsubseteq \forall \text{publicationYear}.\text{Integer}$



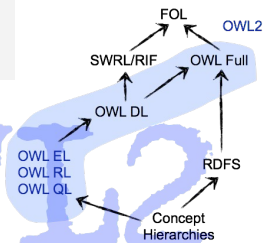
OWL Properties and Individuals

```
:Book a owl:Class .
:Writer a owl:Class .

:author a owl:ObjectProperty ;
    rdfs:domain :Book ;
    rdfs:range :Writer .

:publicationYear a owl:DatatypeProperty ;
    rdfs:domain :Book ;
    rdfs:range xsd:integer .

:GeorgeOrwell a Writer .
:NineteenEightyFour a :Book ;
    :author :GeorgeOrwell ;
    :publicationYear 1948 .
```



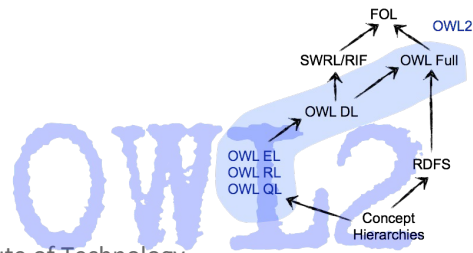
OWL2

OWL Class Hierarchies

```
:Novel a owl:Class ;  
      rdfs:subClassOf :Book .  
:Book a owl:Class ;  
      rdfs:subClassOf :Work .  
:Work a owl:Class .
```

Novel $\sqsubseteq$ Book
Book $\sqsubseteq$ Work

- via inference it can be entailed that :Novel is also a subclass of :Work



HPI Hasso Plattner Institut

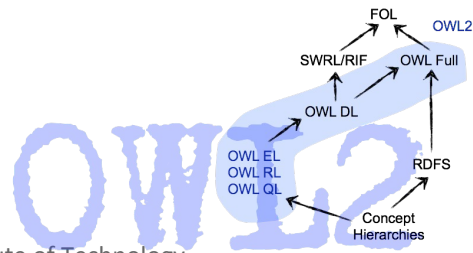
Novel $\sqsubseteq$ Book
Poet $\sqsubseteq$ Writer
Book $\sqcap$ Writer $\sqsubseteq \perp$

-
- Stack of Technology
- OWL2
- OWL EL
OWL RL
OWL QL
- OWL DL
- SWRL/RIF
- FOL
- OWL Full
- RDFS
- Concept Hierarchies
- OWL2

OWL Class Hierarchies and Disjunctiveness

- OWL provides a **shortcut** to define several classes to be disjunctive

```
[ ] a owl:AllDisjointClasses ;  
    owl:members  
      ( :Book  
        :Writer  
        :Vegetable  
        :Furniture  
        :Car ) .
```

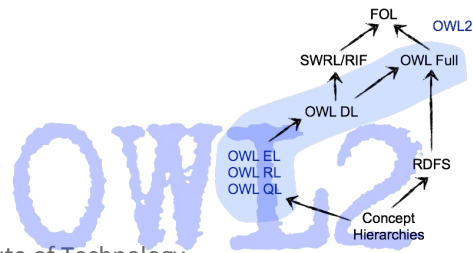


OWL Class Hierarchies and Equivalence

```
:Writer a owl:Class .  
:Author a owl:Class .  
:Poet a owl:Class ;  
      rdfs:subClassOf :Writer .  
  
:Writer owl:equivalentClass :Author .
```

Poet $\sqsubseteq$ Writer
Writer $\equiv$ Author

- via inference it can be entailed that :Poet is also an :Author

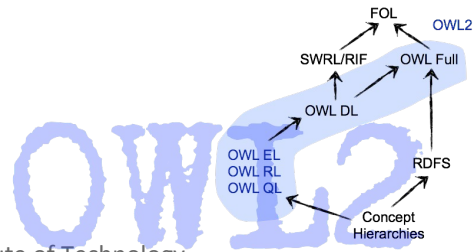


OWL Individuals

```
:NineteenEightyfour a :Novel ;  
    :author :GeorgeOrwell ;  
    :publicationYear 1948 ;  
    owl:sameAs :ARX012345 .  
  
:Novel a owl:Class ;  
    rdfs:subClassOf :Book .  
  
:Book a owl:Class.
```

- via inference it can be entailed that `:ARX012345` is a `:Book`
- difference of Individuals via `owl:differentFrom`

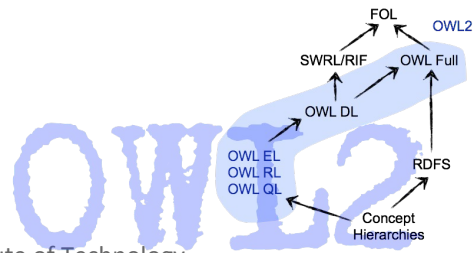
```
:ARX012345 a :Novel ;  
    owl:differentFrom :ARX012346 .
```



OWL Class Hierarchies and Disjunctiveness

- OWL provides a **shortcut** to define several individuals to be different

```
[ ] a owl:AllDifferent ;  
    owl:distinctMembers  
      ( :NineteenEightyfour  
        :AnimalFarm  
        :BraveNewWorld  
        :Simulacron5 ) .
```





Next: 04 - More OWL

Lecture 3: Linked Data Vocabularies and Ontologies - OpenHPI - Course Linked Data Engineering