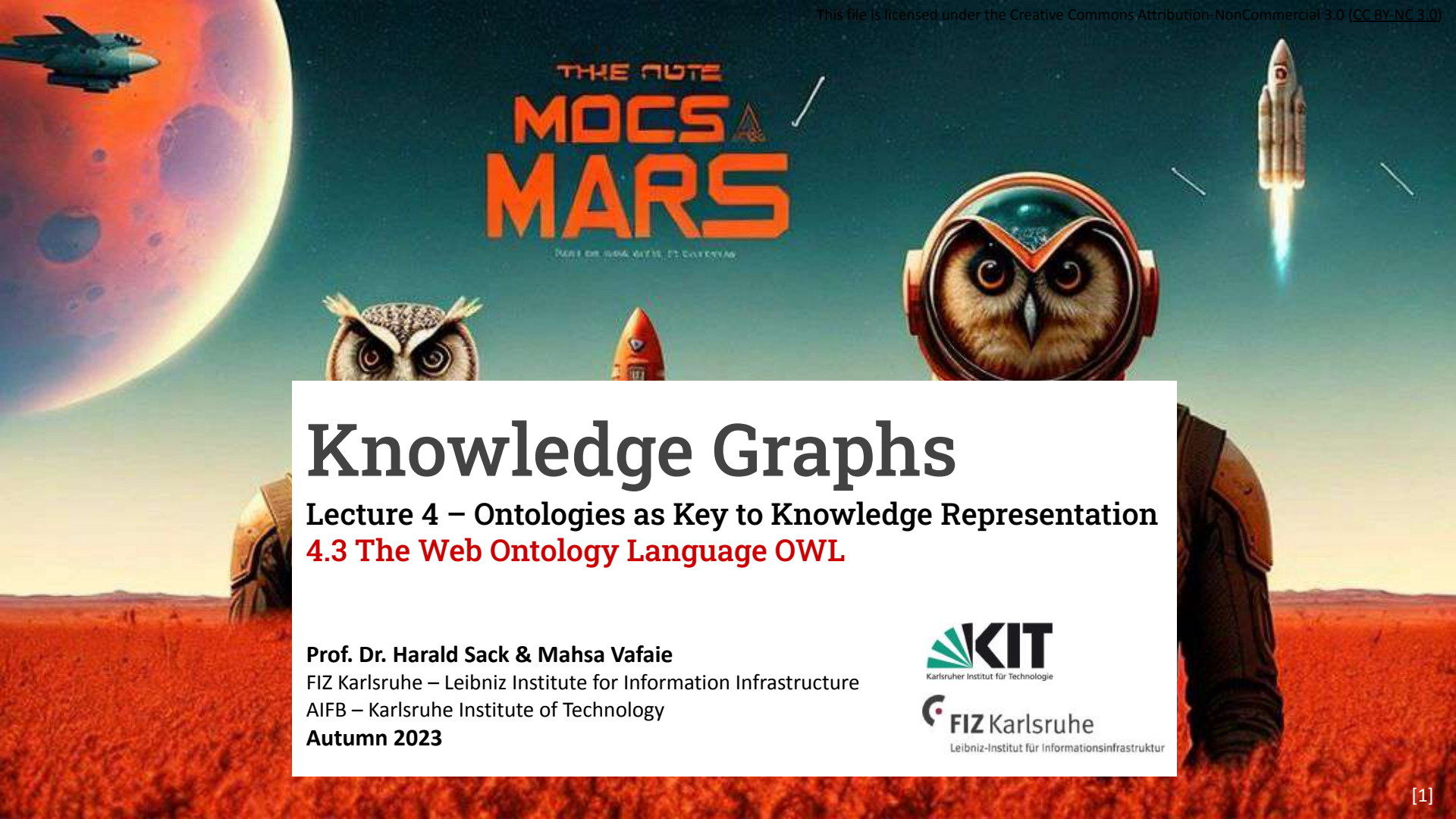


Knowledge Graphs

Lecture 4 – Ontologies as Key to Knowledge Representation

4.3 The Web Ontology Language OWL

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

Karlsruher Institut für Technologie

FIZ Karlsruhe
Leibniz-Institut für Informationsinfrastruktur

Knowledge Graphs

Lecture 4: Ontologies as Key to Knowledge Representation

4.1 From Aristotle to AI: Exploring Ontologies in Computer Science

4.2 The Crucial Role of Mathematical Logic

Excursion 5: Essential Logics in a Nutshell

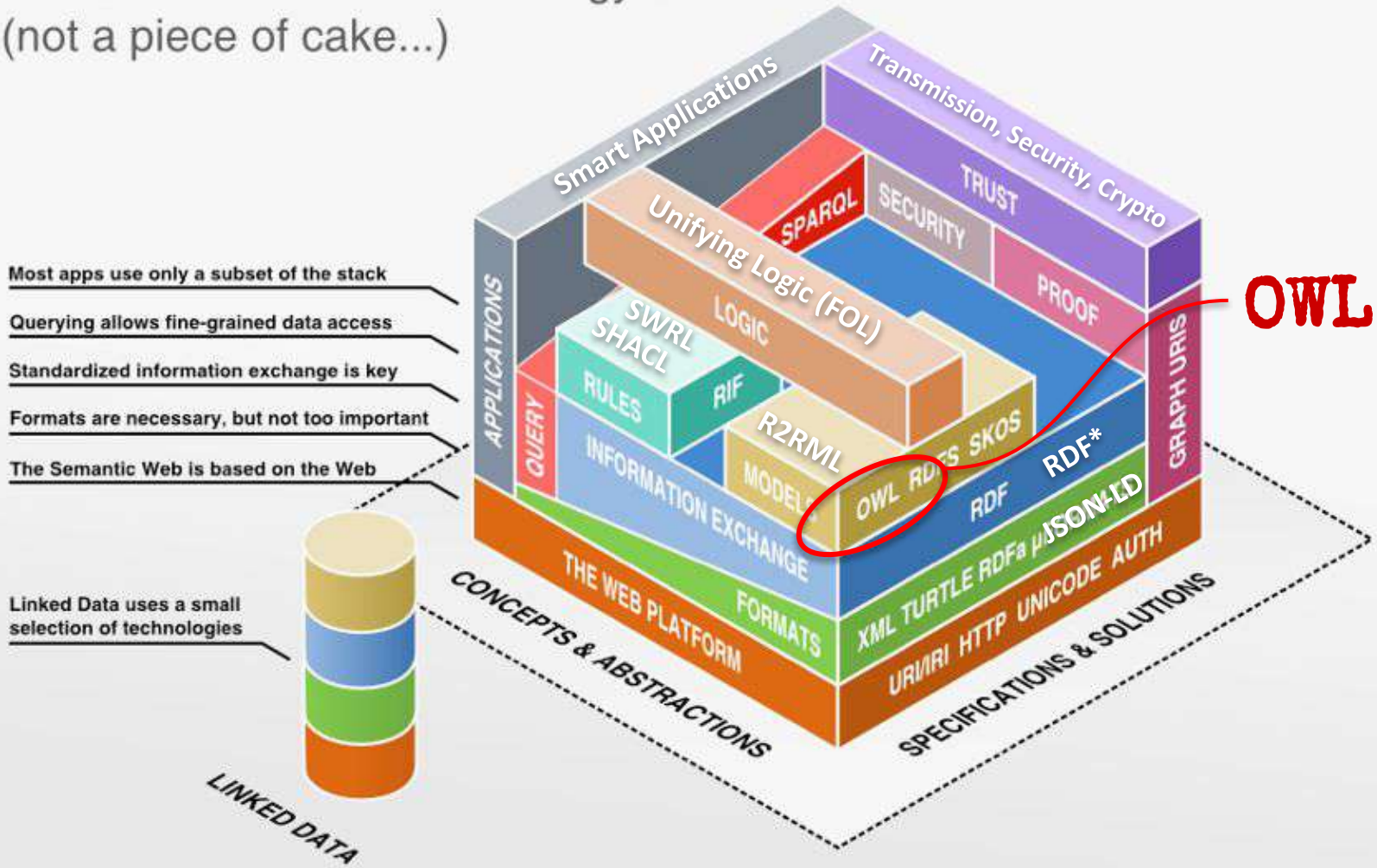
Excursion 6: Description Logics

4.3 The Web Ontology Language OWL

4.4 From simple to complex: Scaling up with OWL

4.5 Unlocking the Potential of OWL

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

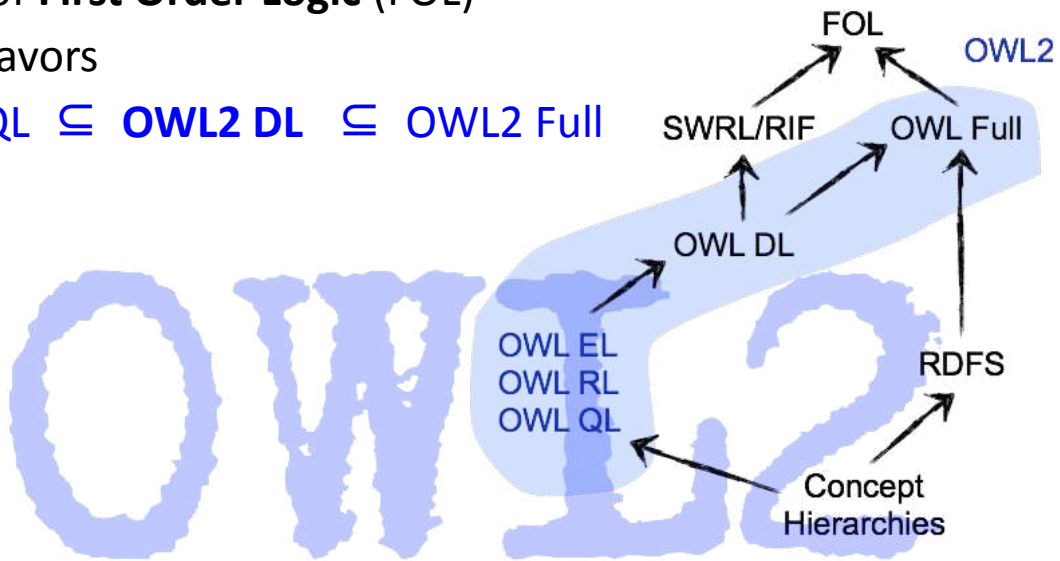


[2]

Web Ontology Language OWL – OWL Flavors

- OWL is a semantic fragment of **First Order Logic** (FOL)
- OWL also exists in different flavors

  OWL EL, OWL RL, OWL QL ⊆ **OWL2 DL** ⊆ OWL2 Full



OWL2 DL is based on 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 $\exists S.Self$
- 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)

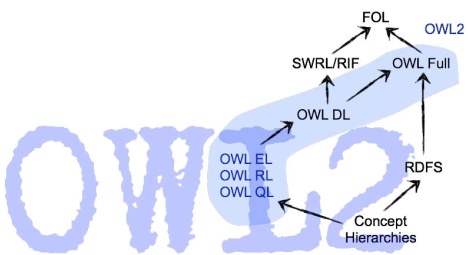
- Inclusion $C \sqsubseteq D$
- Equivalence $C \equiv D$

Rbox (Property Axioms)

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

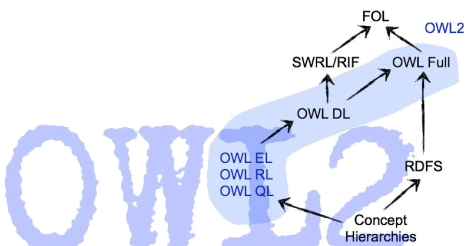
Abox (Facts)

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



OWL2 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



OWL2 Classes

- There exist two **predefined classes**
  - `owl:Thing` (class that contains all individuals)
  - `owl:Nothing` (empty class)

equivalent expression in **description logics**

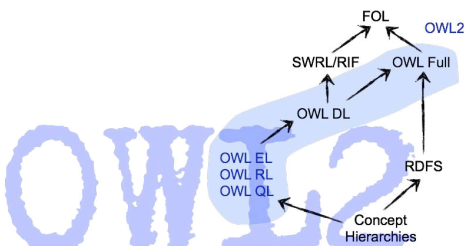
$\top \equiv C \sqcup \neg C$

$\bot \equiv C \sqcap \neg C$

- **Definition of a class**

```
:Person a owl:Class .
```

This is OWL in RDF/Turtle serialization.



OWL2 Individuals

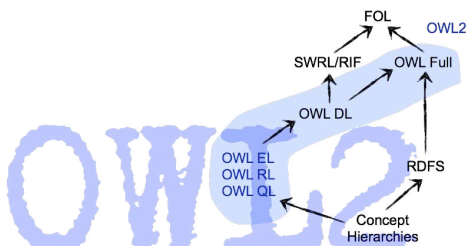
- **Definition of individuals** via class membership

```
:IsaacAsimov a :Person .
```

```
Person(IsaacAsimov)
```

- Individuals can also be defined **without class membership** as **named individuals**

```
:HaraldSack a owl:NamedIndividual .
```



OWL2 Object Properties

- There exist two **property variants**:
  - Object properties
  - Datatype properties

- **Object properties** have classes as range

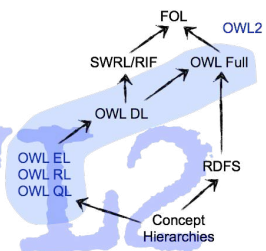
```
:author a owl:ObjectProperty .
```

- **Domain** and **Range** of object properties

```
:author a owl:ObjectProperty ;
        rdfs:domain :Book ;
        rdfs:range  :Person .
```

$\exists$ author.$\top \sqsubseteq$ Book

$\top \sqsubseteq \forall$ author.Person



OWL2 Datatype Properties

- **Datatype properties** have datatypes as range

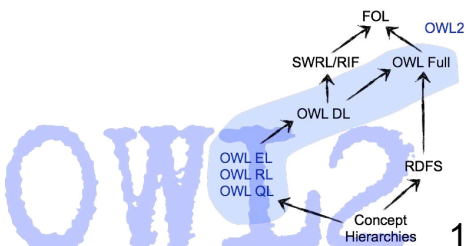
```
:publicationDate a owl:DatatypeProperty .
```

- **Domain** and **Range** of datatype properties

```
:publicationDate a owl:DatatypeProperty ;
                 rdfs:domain owl:Thing ;
                 rdfs:range  xsd:date .
```

$\exists \text{publicationDate}.\top \sqsubseteq \top$

$\top \sqsubseteq \forall \text{publicationDate}.\text{xsd:date}$



OWL2 Properties and Individuals

OWL TBox

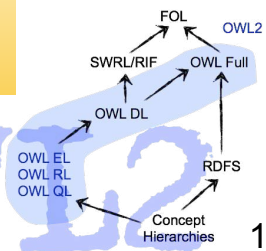
```
:Book a owl:Class .
:Person a owl:Class .

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

:publicationDate a owl:DatatypeProperty ;
                 rdfs:domain owl:Thing ;
                 rdfs:range  xsd:date .
```

```
:IsaacAsimov a Person .
:Foundation a :Book ;
            :author :IsaacAsimov ;
            :publicationDate “1951-08-30”^^xsd:date .
```

OWL ABox



OWL2 Class Hierarchies

```
:Poet a owl:Class ;
      rdfs:subClassOf :Writer .
:Writer a owl:Class ;
      rdfs:subClassOf :Person .
:Person a owl:Class .
```

we don't need to define a new subClassOf property for owl, we simply reuse rdfs:subClassOf

$$\text{Poet} \sqsubseteq \text{Writer}$$
$$\text{Writer} \sqsubseteq \text{Person}$$

Via **inference** it can be entailed that `:Poet` is also a subclass of `:Person`.

$$\left.\begin{array}{l}\text{Poet} \sqsubseteq \text{Writer}\\ \text{Writer} \sqsubseteq \text{Person}\end{array}\right\} \models \text{Poet} \sqsubseteq \text{Person}$$

OWL2 Class Hierarchies and Disjunctiveness

```
:Book a owl:Class .
:Person a owl:Class .
:Novel a owl:Class ;
       rdfs:subClassOf :Book .
:Author a owl:Class ;
      rdfs:subClassOf :Person .

:Book owl:disjointWith :Person .
```

In OWL everything might be potentially identical if we don't explicitly state the difference.

$$\begin{aligned} &\texttt{Novel} \sqsubseteq \texttt{Book} \\ &\texttt{Author} \sqsubseteq \texttt{Person} \\ &\texttt{Book} \sqcap \texttt{Person} \sqsubseteq \bot \end{aligned}$$

Via **inference** it can be entailed that `:Novel` and `:Author` are also disjoint classes.

$$\left.\begin{aligned} &\texttt{Novel} \sqsubseteq \texttt{Book} \\ &\texttt{Author} \sqsubseteq \texttt{Person} \\ &\texttt{Book} \sqcap \texttt{Person} \sqsubseteq \bot \end{aligned}\right\} \models \texttt{Novel} \sqcap \texttt{Author} \sqsubseteq \bot$$

OWL2 Class Hierarchies and Equivalence

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

:Author owl:equivalentClass :Writer .
```

$$\texttt{Poet} \sqsubseteq \texttt{Writer}$$
$$\texttt{Writer} \equiv \texttt{Author}$$

Via **inference** it can be entailed that `:Poet` is also an `:Author`.

$$\left.\begin{array}{l}\texttt{Poet} \sqsubseteq \texttt{Writer}\\ \texttt{Writer} \equiv \texttt{Author}\end{array}\right\} \models \texttt{Poet} \sqsubseteq \texttt{Author}$$

OWL2 Individuals – Identity and Distinctiveness

```
:Foundation a :Novel ;
    :author :IsaacAsimov ;
    :publishingDate "1951-08-30"^^xsd:date ;
    owl:sameAs :ARX012345 .

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

:Book a owl:Class.
```

For identical individuals: owl:sameAs
For identical classes: owl:equivalentClass

- Via **inference** it can be entailed that :ARX012345 is a :Book.
- **Difference of Individuals** via owl:differentFrom.

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


From simple to complex:
Scaling up with OWL
Next Lecture...

Bibliographic References:

- Pascal Hitzler, Markus Krötzsch, Bijan Parsia, Peter F. Patel-Schneider, Sebastian Rudolph (eds., 2012), *OWL 2 Web Ontology Language Primer (Second Edition)*, W3C Recommendation 11 December 2012
- Aidan Hogan (2020), *The Web of Data*, Springer.
  Chap. 5.4 OWL 2 Vocabulary, 196–242.

Picture References:

[1] "A Scifi movie poster "The Owls of Mars" depicting a huge owl sitting in the lonely red prairies of Mars in a retro futuristic rural environment of planet Mars. A rocket ship is starting in the background far away leaving contrails behind.", 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] "Several owls are walking on a crowded street in a Bladerunner like dystopian city environment.", created via ArtBot, Deliberate, 2023, [CC-BY-4.0], https://tinybots.net/artbot