

Een introductie in het kader van toepassingen binnen de gebouwde omgeving

## Introductie tot LinkedData



Sheets alvast ophalen voor op eigen scherm  
(DigiGo – Training LinkedData – Slides.pdf):

**[t.ly/4vg9f](https://t.ly/4vg9f)**

! Hoofdletter gevoelig

# Rik



# Praktische kennis, direct toepasbaar

Kennisplatform CROW ontwikkelt en verrijkt collectieve kennis over **infrastructuur** en **mobiliteit**.

Als kennisplatform ontwikkelen we **praktische oplossingen** en bevorderen we de **toepasbaarheid** van deze kennis voor én met de sector.

Zo is CROW de drijvende kracht achter een **duurzame inrichting** van de fysieke leefomgeving in Nederland.

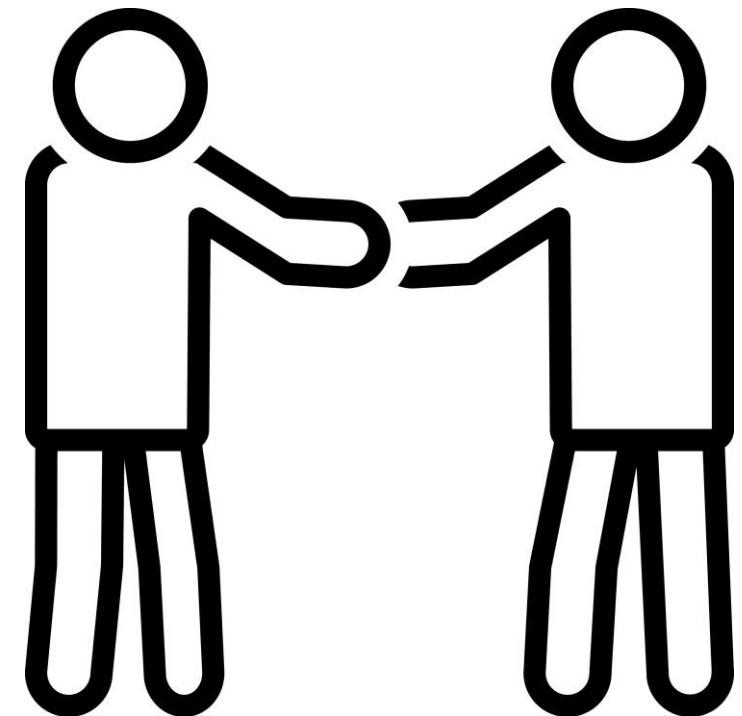
*“Iedere Nederlander ondervindt dagelijks de impact van ons werk.”*

# Programma van vandaag

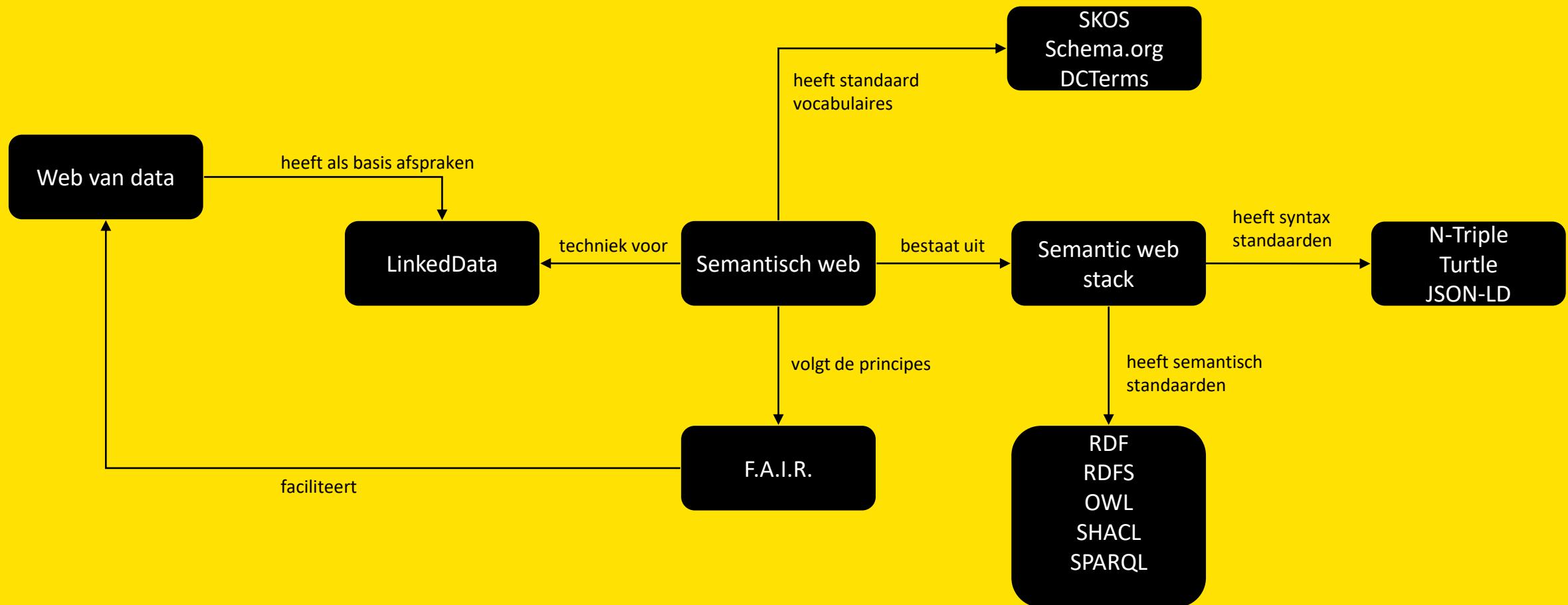
- 10:00 – 10:15: Voorstellen & Verwachtingen
- 10:15 – 11:00: *LinkedData ‘wat is het’*: FAIR, LinkedData en het Semantisch web
- Korte pauze
- 11:15 – 12:15: *LinkedData ‘hoe maak je het’*: Semantisch modelleren, NEN2660-2
- Lunch
- 13:15 – 14:15: *LinkedData ‘hoe gebruik je het’*: SPARQL en cases
- Korte pauze
- 14:30 – 15:30 (- 16:00): *LinkedData ‘hoe gebruik je het’*: Tools, oefeningen
- Evaluatie en borrel

# Voorstellen

- Naam, organisatie, functie
- Ervaring met LinkedData
- Je verwachting van vandaag (aandacht voor, doel, ...)
- *Wat levert het jouw organisatie op als jij dit weet*



# Model van de training



# LinkedData: wat is het

**Van document-gebaseerd Web, naar een Web van onderling verbonden data**



Er is (te) veel informatie voor de mens om te verwerken



Maar computers willen alles simplificeren naar 1 en 0.

# F.A.I.R. doel

De principes benadrukken **machine-actie**. Ofwel de mogelijk van computers om data met weinig of geen menselijke tussenkomst:

- te vinden
- toegang te krijgen
- te laten samenwerken
- en opnieuw te gebruiken

Wij vertrouwen steeds meer op ondersteuning van computers om met data om te gaan als gevolg van de **toename** in volume, **complexiteit** en **snelheid** van data-creatie.

De principes verwijzen naar drie soorten dingen: data ('een digitaal object'), metadata (informatie over dat digitale object) en (ict)infrastructuur.

# GO FAIR principles

## Findable / Vindbaar

(Meta)data moet makkelijk te vinden zijn voor zowel mens en computer. Machine leesbare metadata zijn essentieel voor automatische ontdekking van datasets.

- F1. (Meta)data hebben een wereldwijd uniek en blijvend identificatienummer.
- F2. Data worden beschreven met rijke metadata (zie R1).
- F3. Metadata omvatten duidelijk en expliciet het identificatienummer van de data die ze beschrijven.
- F4. (Meta)data worden geregistreerd of geïndexeerd in een doorzoekbare bron.

## Accessibel / Toegankelijk

Als de gebruiker de benodigde data heeft gevonden, moet men weten hoe deze kan worden benaderd, mogelijk inclusief authenticatie en autorisatie.

- A1. (Meta)data zijn op te halen via hun identificatienummer met behulp van een gestandaardiseerd communicatieprotocol.
  - A1.1 Het protocol is open, gratis en universeel implementeerbaar.
  - A1.2 Het protocol maakt, indien nodig, een authenticatie- en autorisatieprocedure mogelijk.
- A2. Metadata zijn toegankelijk, zelfs wanneer de data niet langer beschikbaar zijn.

# GOFAIR principles

## Interoperable / Interoperabel

Data moet meestal worden geïntegreerd met andere data. Bovendien moeten de data samenwerken met toepassingen voor analyse, opslag en verwerking.

- I1. (Meta)data gebruiken een formele, toegankelijke, gedeelde en breed toepasbare taal voor kennisrepresentatie.
- I2. (Meta)data gebruiken woordenschatten die de FAIR-principes volgen.
- I3. (Meta)data bevatten gekwalificeerde verwijzingen naar andere (meta)data.

## Fully AI Ready

## Reusable / Herbruikbaar

Het uiteindelijke doel van FAIR is hergebruik van data. (Meta)data moet goed beschreven zijn zodat ze het kan worden gerepliceerd en/of gecombineerd in verschillende omgevingen.

- R1. (Meta)data worden rijk beschreven met een veelheid aan nauwkeurige en relevante attributen.
- R1.1. (Meta)data worden vrijgegeven met een duidelijke en toegankelijke data-gebruik licentie.
- R1.2. (Meta)data zijn geassocieerd met gedetailleerde herkomstinformatie.
- R1.3. (Meta)data voldoen aan domeinrelevante gemeenschapstandaarden.

## Web van documenten

- Sinds de jaren '90
- Link (URL) voor elke pagina
- Een site is een document
- Links naar anders pagina's
- Voor mensen

# de Volkskrant

Voorpagina Columns Opinie Cartoons Cultuur & Media Podcasts Foto Puzzels



**BELLEN MET** ONZE POLITIEK FOTOGRAAF Veertien jaar Rutte vastleggen: 'Je wordt onderdeel van het politieke theater'



**ANALYSE**  
Oranje is niet meegegroeid met de moderne voetbalcultuur



Zoeken Edite Instellingen

Abonneren vanaf €2,25 per week

Log in

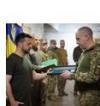
NIEUW  
Abonneer nu op de Volkskrant  
Al vanaf 2,25 per week

→

**NIEUWS**  
■ 10:15 UUR  
Live EK: Roemenië moet andere linksback opstellen tegen Nederland, Bancu geschorst na tweede gele kaart



■ 09:30 UUR  
Live Midden-Oosten: Eerste kandidaat presidentsverkiezingen Iran trekt zich terug



■ 07:59 UUR  
Live Oekraïne: Europese Unie: herinvoering belasting op Oekraïense eieren en suiker



■ 05:00 UUR  
De radio blijft onverminderd populair, maar dankt die reputatie wel aan oudere luisteraar



■ 03:30 UUR  
Oud-president Honduras in de Verenigde Staten veroordeeld tot 45 jaar cel voor drugshandel

## Gaat prima toch?

- Semantiek van pagina is begrijpelijk
- Als we taal, schrift en domein weten

Maar:

- Semantiek is moeilijk voor machines
- Context mist vaak
- Voorbeeld: Jaguar



## Sheet uit 2020: Linked Data and Semantic Web: Lecture 3 van Universiteit Twente

# KNOWLEDGE GRAPHS AND RDF TODAY

**Hype Cycle for Artificial Intelligence, 2020**

The graph illustrates the Hype Cycle for Artificial Intelligence in 2020. The Y-axis represents 'Expectations' and the X-axis represents 'Time' with five distinct phases: Innovation Trigger, Peak of Inflated Expectations, Trough of Disillusionment, Slope of Enlightenment, and Plateau of Productivity. Technologies are plotted along the curve, with some marked as obsolete before the plateau. Knowledge Graphs are highlighted and circled.

Technology	Time to Plateau
Knowledge Graphs	less than 2 years
Intelligent Applications	2 to 5 years
Deep Neural Network ASICs	5 to 10 years
Data Labeling and Annotation Services	more than 10 years
Smart Robots	more than 10 years
Decision Intelligence	more than 10 years
AI Developer and Teaching Kits	more than 10 years
AI Governance	more than 10 years
Augmented Intelligence	more than 10 years
Neuromorphic Hardware	more than 10 years
Things as Customers	more than 10 years
Responsible AI	more than 10 years
AI Marketplaces	more than 10 years
Small Data	more than 10 years
Artificial General Intelligence	more than 10 years
Generative AI	more than 10 years
Composite AI	more than 10 years
FPGA Accelerators	more than 10 years
Chatbots	more than 10 years
Machine Learning	more than 10 years
Digital Ethics	more than 10 years
Edge AI	more than 10 years
AI Cloud Services	more than 10 years
Deep Neural Networks (Deep Learning)	more than 10 years
Natural Language Processing (NLP)	more than 10 years
Computer Vision	more than 10 years
Autonomous Vehicles	more than 10 years
Cognitive Computing	more than 10 years
Insight Engines	more than 10 years
GPU Accelerators	more than 10 years

Plateau will be reached:

- less than 2 years
- 2 to 5 years
- 5 to 10 years
- ▲ more than 10 years
- ✖ obsolete before plateau

As of July 2020

Industry (2020): “*we need to use knowledge graphs!*”

→ Nice name for an *old idea*... In 1993 there were people already using the term “ontology” for *data representation*!

→ How *Google* call it nowadays?

*Google Structured Data*

→ Who else in *industry* uses these technologies (*in production*)? In academia: everywhere...

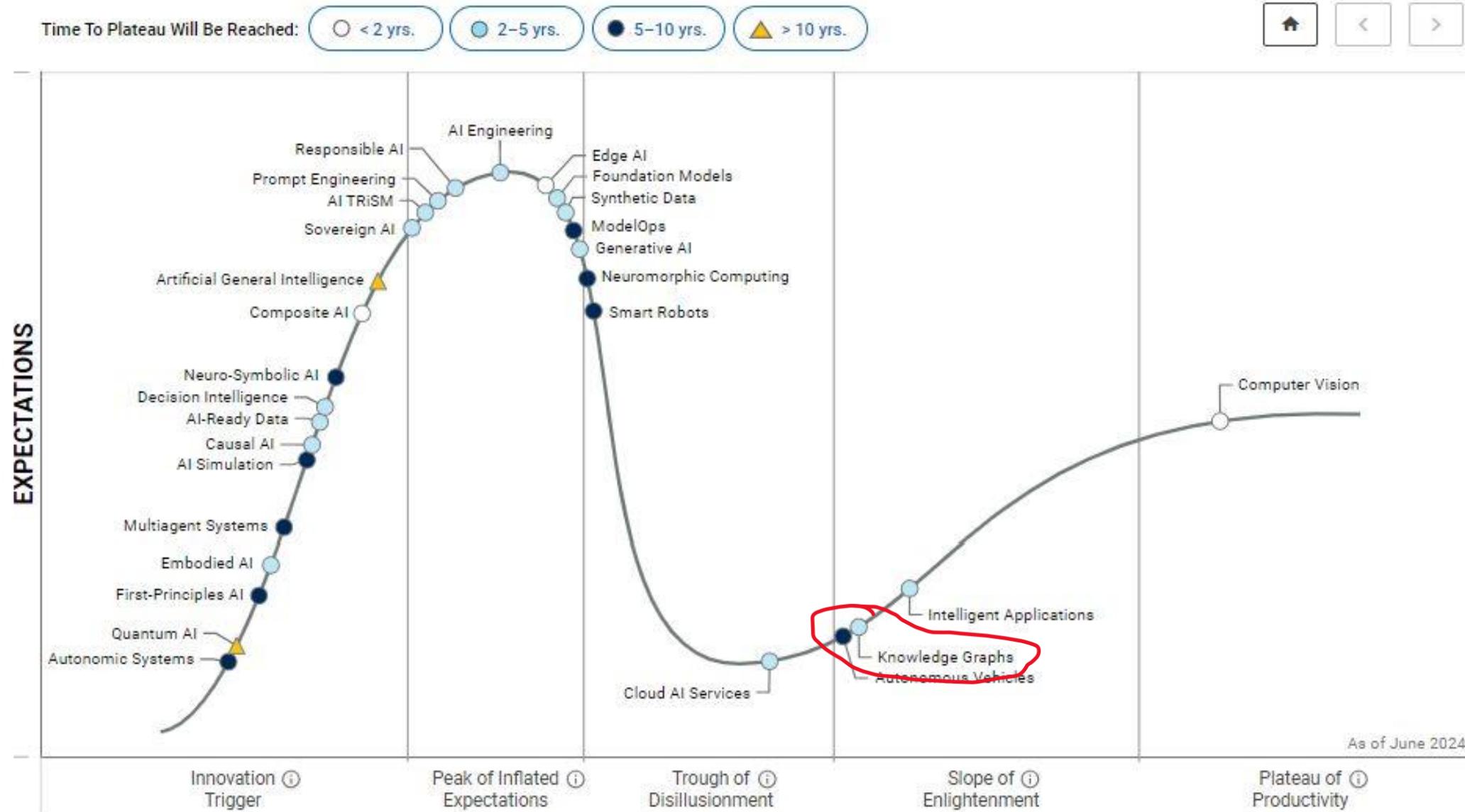
*Fraunhofer*

... and many many others....

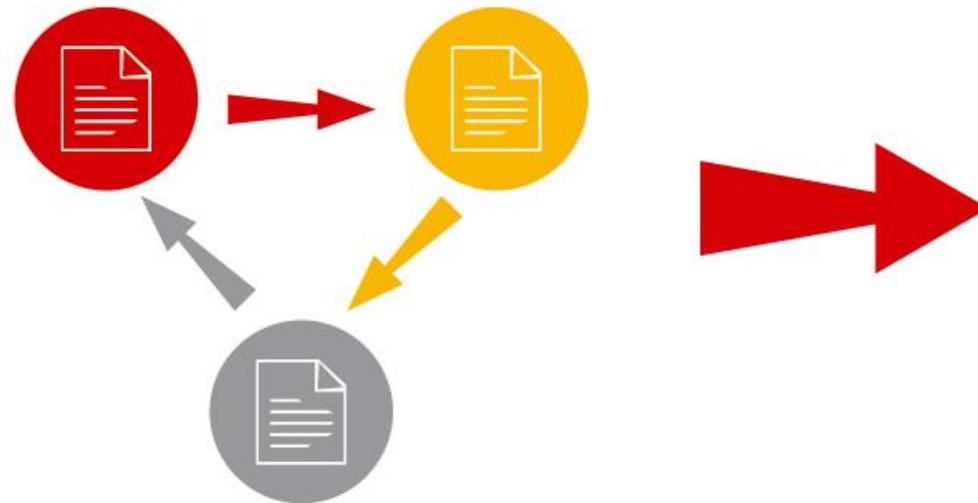
→ SW technologies: already pointed as the *future of data science*

UNIVERSITY  
OF TWENTE.

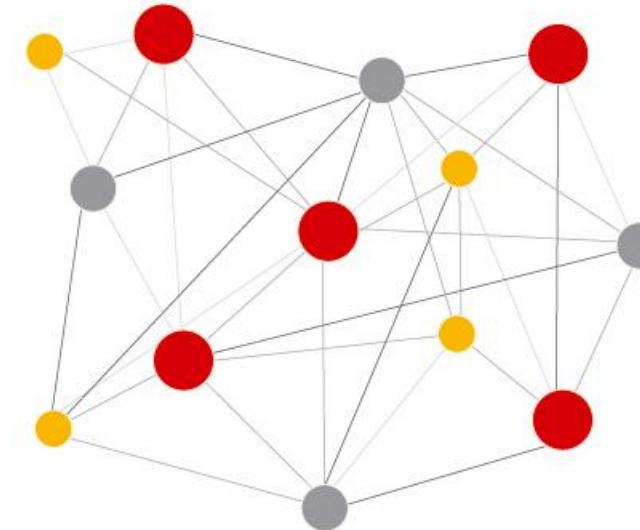
<https://towardsdatascience.com/knowledge-data-science-with-semantics-technologies-ff54e4fe306c>



## Web van documenten (HTML)



## Web van data (RDF)



*Set van best practices om gestructureerde gegevens op het web te publiceren en te verbinden*

*Tim Berners Lee: The next web (TED2009)*

[https://www.ted.com/talks/tim\\_berners\\_lee\\_the\\_next\\_web](https://www.ted.com/talks/tim_berners_lee_the_next_web)

# Gebruik van LinkedData

- IT, Zorg, Farmacie, Overheid en Wetenschap
- Bedrijfsintern, voor informatiedeling om bedrijfsprocessen op elkaar aan te laten sluiten
- Nederlandse voorbeelden: Kadaster, Rijkswaterstaat, Schiphol, ProRail, Arcadis, Besix, Ministerie van Defensie, Politie, uitgeverij Elsevier, CROW.

Mooiste, bekendste en grootste bron: [wikidata.org](https://www.wikidata.org)



Welcome!

Wikidata is a free and open knowledge base that can be read and edited by both humans and machines.

Wikidata acts as central storage for the **structured data** of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others.

Wikidata also provides support to many other sites and services beyond just Wikimedia projects! The content of Wikidata is available under a free license, exported using standard formats, and can be interlinked to other open data sets on the linked data web.

# Vuistregels voor verschillen

- Relationale databases
- *Tabellen met records en kolommen*
- *Vast gedefinieerde structuur (schema per database)*
- Triplestore
- *'1 tabel met drie kolommen'*
- *Schema-loos (zelfbeschrijvend)*
  
- REST-API
- *Vooraf gedefinieerde vragen*
- *Specifieke toepassingen*
- SPARQL-Endpoint
- *Query taal*
- *Alle vragen mogelijk*



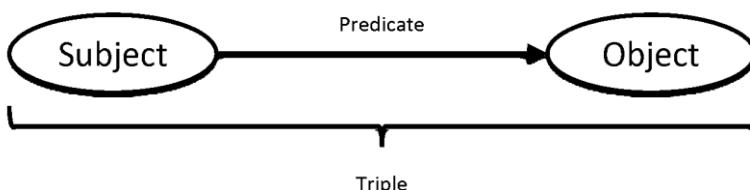
# Voor en Na

- **VOOR LinkedData**
  - *waar staat je informatie*
  - *hoe lees ik het uit*
  - *waar staat wat ik wil weten*
  - *hoe moet ik het interpreteren*
  - *hoe moet de computer het interpreteren*
- **NA LinkedData**
  - *waar staat de informatie*
  - *wat wil ik weten*



## RDF

- Het datamodel voor LinkedData en Semantisch web
- Meest simpele vorm van informatie opslag
- **Resource:** pagina's, concepten, ideeën, ... (alles wat een URI kan hebben)
- **Description:** attributen, kenmerken en relaties van de resources
- **Framework:** model, taal en syntax voor deze descriptions
- Relateert twee dingen (subject en object), middels een relatie (predicaat)



## URI

- **Unique, resource, identifier**
- De unieke ID van een resource
- Elk datapunt heeft een uniek ID in de vorm van een (bezoekbare) URL
- HTTP URI's (URL) zorgen voor menselijke leesbaarheid
- Machines en mensen weten zo het onderscheidt
- Combinatie van **namespace** + '**local ID**'

- <https://data.crow.nl/imbor/def/83a942f7-5291-42f0-afb1-9a57d0fb2f15>
- <https://bgt.basisregistraties.overheid.nl/bgt2/def/Aquaduct>

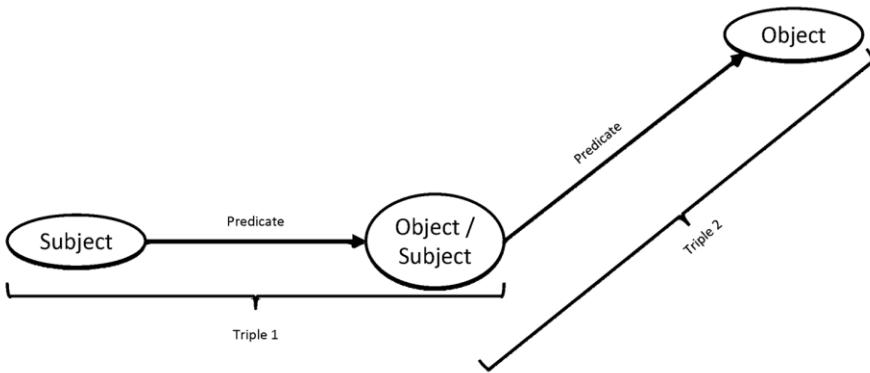
## Taal onafhankelijk

- Machine taal is RDF
- Menselijke taal is geregeld via labels
- Labels voor menselijke leesbaarheid, maar onafhankelijk van de URI
- De informatie is dus taalgebaseerd, maar niet afhankelijk van taal

<http://publications.europa.eu/resource/authority/country/NLD>

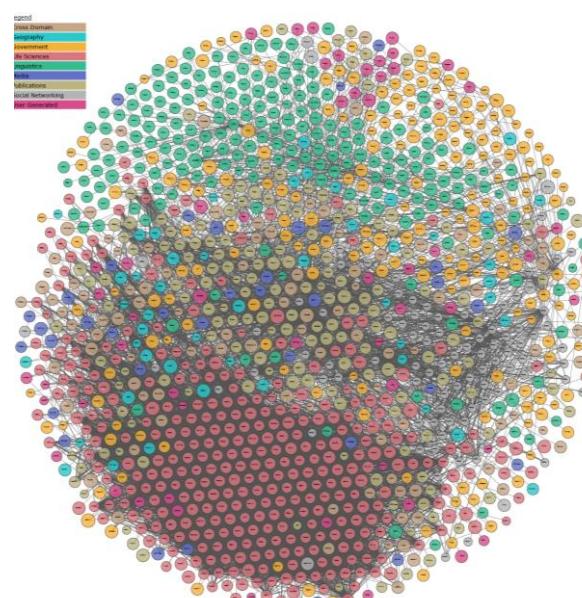
## Linkbaar

- Informatie staat in triples, en kan daardoor hergebruikt worden
- Soms is iets het subject, soms is iets het object
- RDF Graph: setjes of subject-predicate-object triples
- Machines kunnen het hele pad volgens middels de URI's en zo ontstaat een web van data



## Flexibel

- Door gebruik van triple en URI
- Nieuwe data kan eenvoudig bijgevoegd worden zonder bestaande data te veranderen



## Open standaard

- W3C standaard
- Data publiceren op het (semantisch) web
- Hergebruik van informatie, refereren naar andere bronnen
- Mens en machines kunnen context snappen door de links te volgen



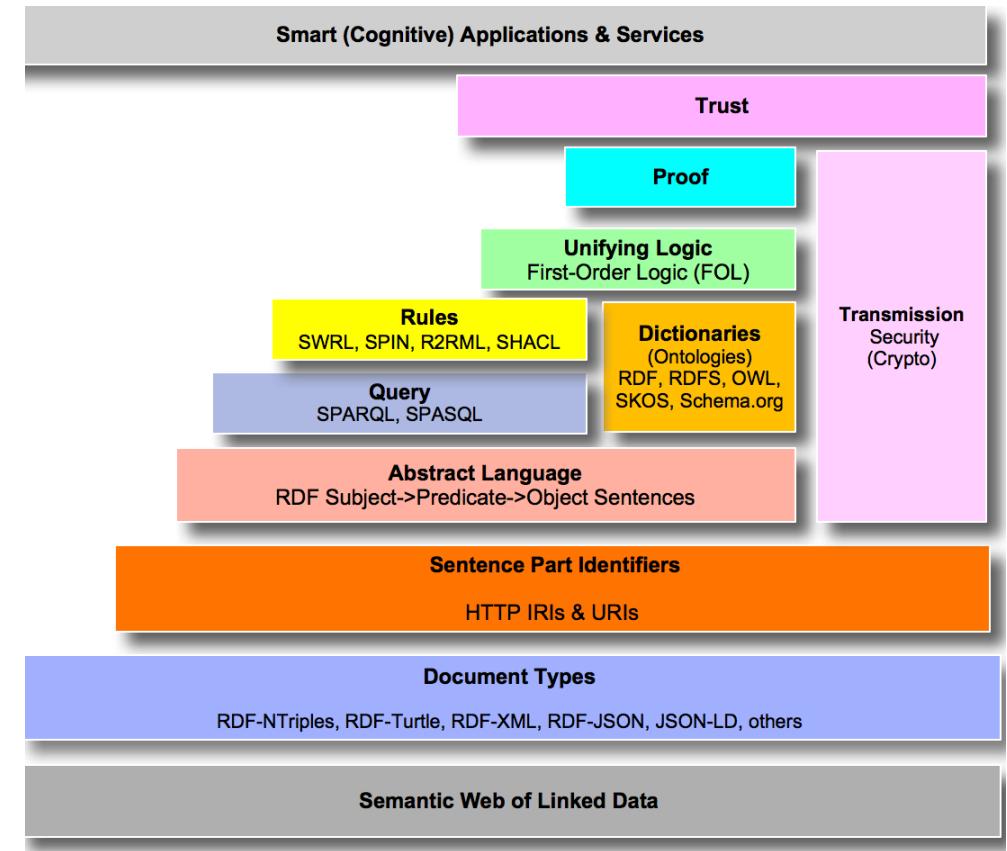


# Semantisch web

Geeft betekenis aan datgene dat in webpagina's staat en legt hier relaties tussen. Het gaat daarmee een stuk verder dan het eerste web dat 'slechts' bestaat uit een verzameling van webpagina's.

Ontwikkeld door een groep mensen van het W3C. Met het idee om betekenisvolle informatie tussen machines uit te wisselen zonder menselijke tussenkomst. Machineleesbaarheid zorgt dat er automatisch relaties binnen en tussen documenten gelegd worden.

Daarvoor moeten op diverse niveaus of lagen standaarden en protocollen afgesproken worden, die W3C hiervoor ontwikkeld heeft.

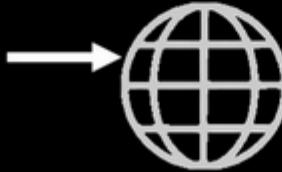


# Semantisch web

**URL → URI**

identify what exists on the web

<http://my-site.fr>

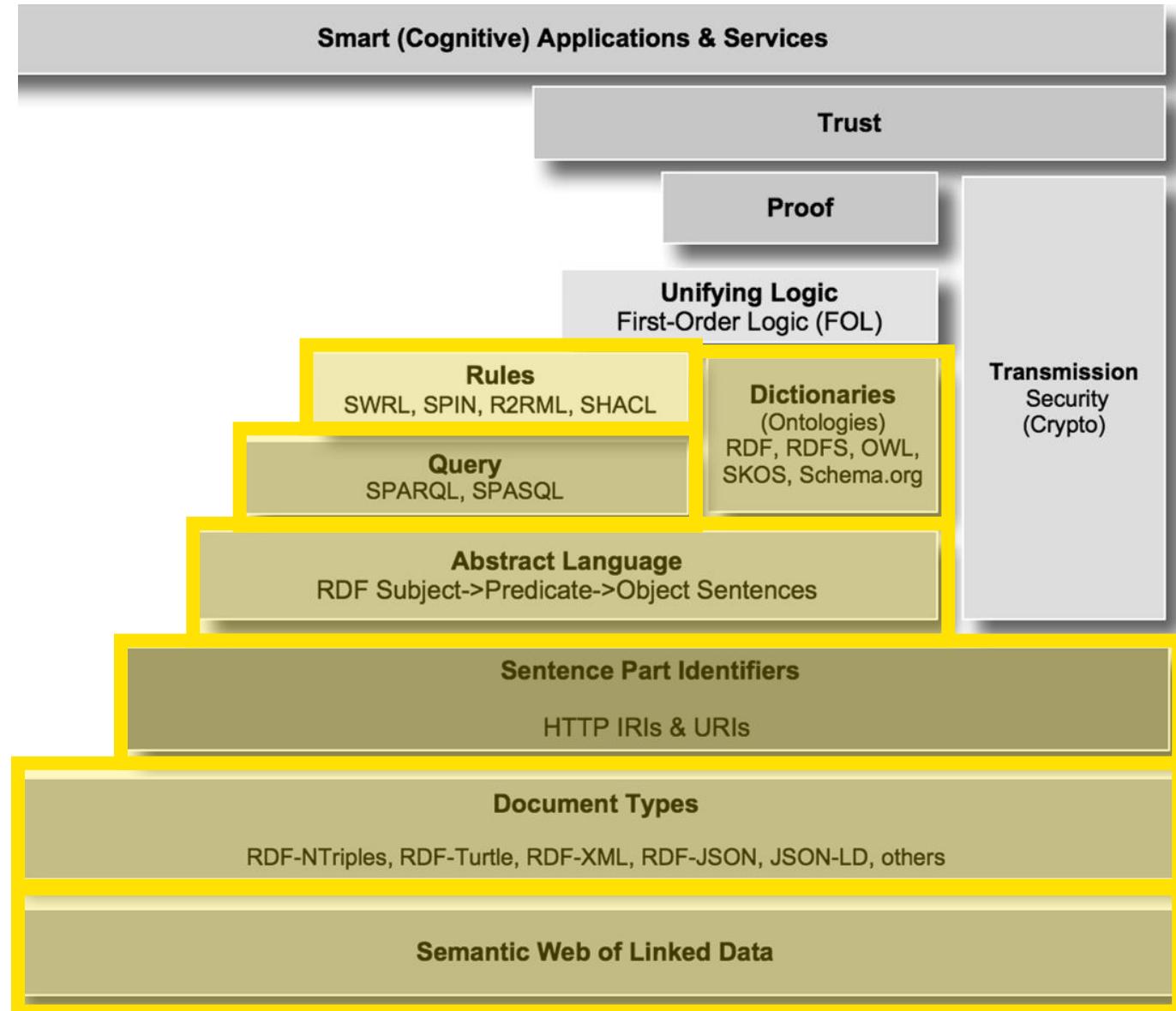


identify, on the web, what exists

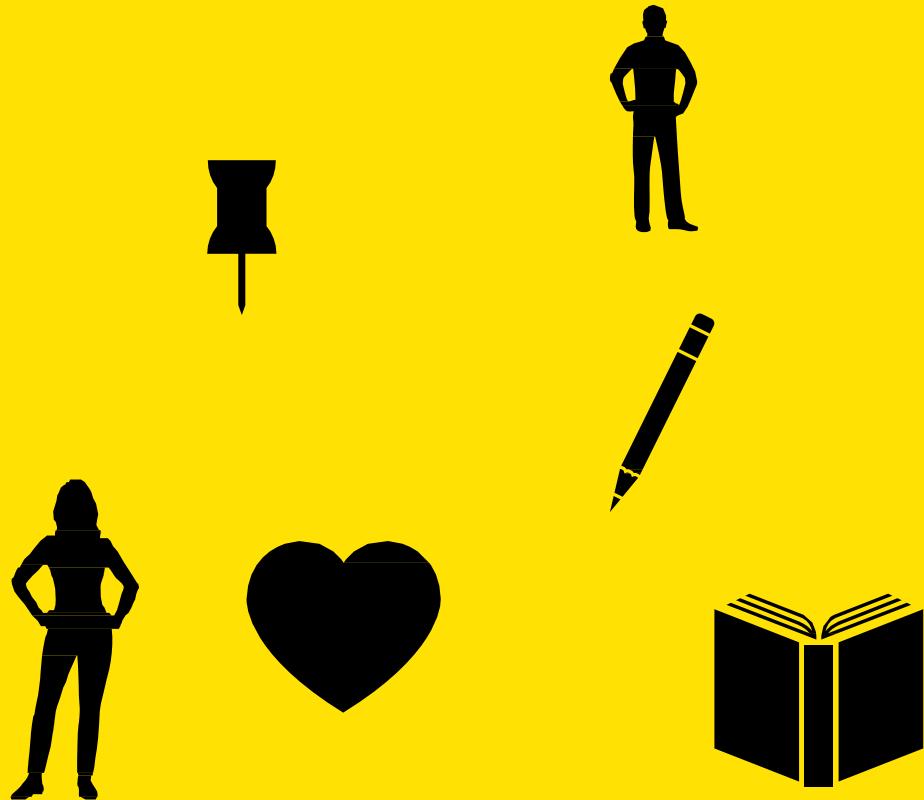
<http://animals.org/this-zebra>



# Semantic web stack

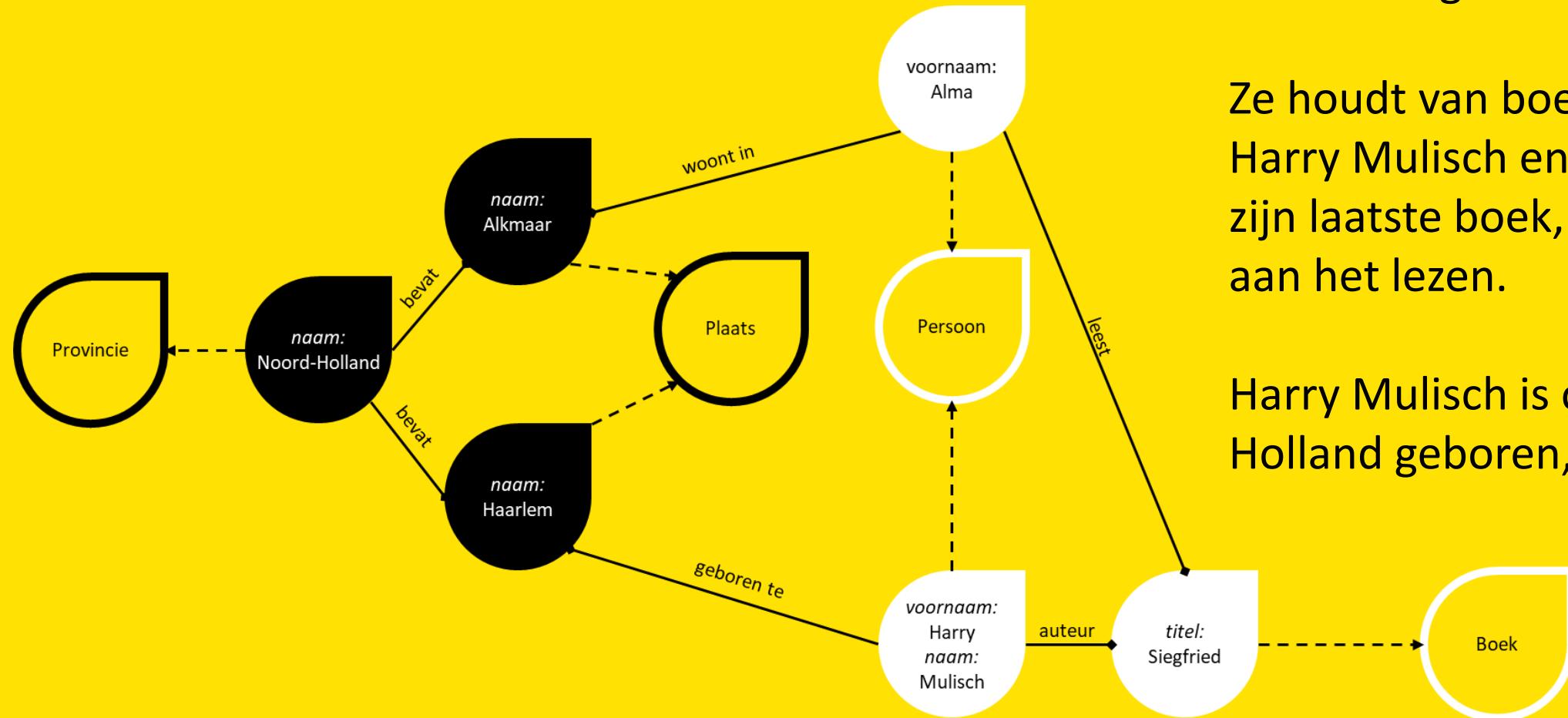


# Semantiek



We kennen Alma. Ze is woonachtig in Alkmaar. Ze houdt van boeken van Harry Mulisch en ze is nu aan zijn laatste boek, Siegfried, aan het lezen. Harry Mulisch is ook in Noord-Holland geboren, in Haarlem.

# Semantisch model

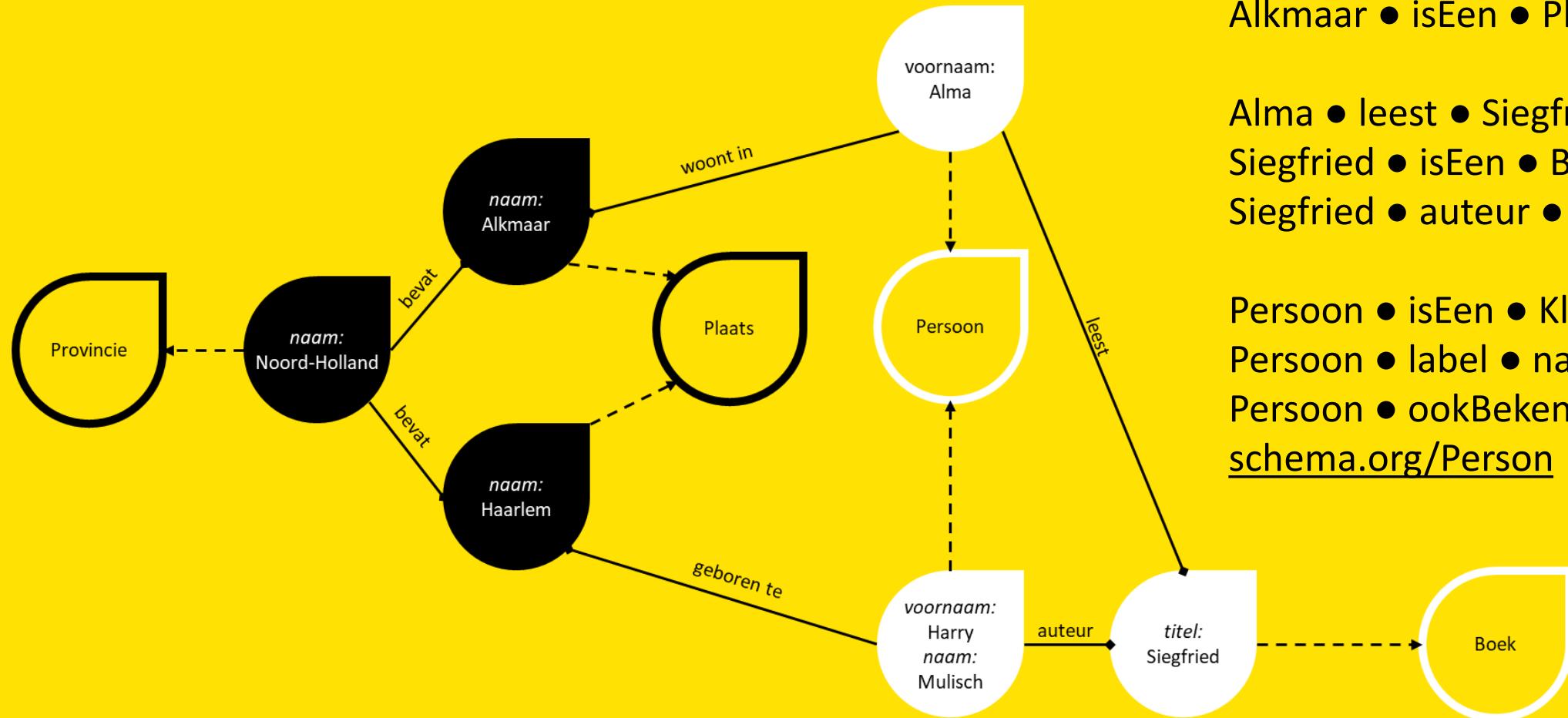


We kennen Alma. Ze is woonachtig in Alkmaar.

Ze houdt van boeken van Harry Mulisch en ze is nu aan zijn laatste boek, Siegfried, aan het lezen.

Harry Mulisch is ook in Noord-Holland geboren, in Haarlem.

# Triples



Alma • isEen • Persoon  
Alma • woontIn • Alkmaar  
Alkmaar • isEen • Plaats

Alma • leest • Siegfried  
Siegfried • isEen • Boek  
Siegfried • auteur • Harry Mulisch

Persoon • isEen • Klasse  
Persoon • label • naam  
Persoon • ookBekendAls • schema.org/Person

# RDF <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

## Inhoud

- Classes:
  - rdf:Property, rdf:Statement, rdf:XMLLiteral
  - rdf:Seq, rdf:Bag, rdf:Alt, rdf>List
- Properties
  - rdf:type, rdf:subject, rdf:predicate, rdf:object
  - rdf:first, rdf:rest, rdf:\_n
  - rdf:value
- Resources:
  - rdf:nil

## Inhoud

- Typeringen:
  - <A, rdf:type, B>  
“A behoort bij class B”
- Alle properties zijn van één class:
  - <P, rdf:type, rdf:Property>  
“P is een property”
  - <rdf:type, rdf:type, rdf:Property>  
“rdf:type is een property”



# RDF <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

## Literals

- Plain literals:  
"hallo wereld"
- Optionele language tag:  
"Hello world?"@en-GB
- Typed literals  
"hello"^^xsd:string, "1"^^xsd:integer
- Gebruikte datatypes:  
XML Schema datatypes
- Alleen bij *object* van een triple:  

```
<http://example.org#henk>
  <http://example.org#hasName>
    "Henk Smit"^^xsd:string
```

## Datatypen

- Datatypes van XML Schema:  
xsd:string  
xsd:integer  
xsd:decimal  
xsd:float  
xsd:anyURI  
xsd:boolean  
xsd:dateTime

## Blank nodes

- Nodes zonder URI
- Resources zonder naam
- More complex constructs
- Notatie van blank nodes hangt van de syntax af

```
<#henk> <#hasName> _:henksnaam
_:henksnaam <#firstName> "Henk"^^xsd:string
_:henksnaam <#lastName> "Smit"^^xsd:string
```

# RDFS <http://www.w3.org/2000/01/rdf-schema#>

## Resource

- Alles is (impliciet) een instantie van:  
rdfs:Resource

## Class

- Beschrijven een set van resources
- Elke class is (expliciet) lid van:  
rdfs:Class
- Classes zijn zelf resources, zoals webpagina's, personen of document typen
- Class hierarchie wordt aangegeven door:  
rdfs:subClassOf

## Property

- Subset van rdfs:Resource, maar zijn properties
- Domain: class waar de property aan zit:  
rdfs:domain
- Range: type van de mogelijke property waarde:  
rdfs:range
- Property hierarchie wordt aangegeven door:  
rdfs:subPropertyOf

# RDFS <http://www.w3.org/2000/01/rdf-schema#>

## RDF Schema

Taal voor definiëren van RDF typen:

- Definiér classes:  
“#Student is een class”
- Relaties tussen classes:  
“#Student is een sub-class van #Persoon”
- Properties van classes:  
“#Persoon heeft een property hasName”

## Inhoud (extensie RDF)

- RDFS Classes
  - rdfs:Resource
  - rdfs:Class
  - rdfs:Literal
  - rdfs:Datatype
  - rdfs:Container
  - rdfs:ContainerMembershipProperty
- RDFS Properties
  - rdfs:domain
  - rdfs:range
  - rdfs:subPropertyOf
  - rdfs:subClassOf
  - rdfs:member
  - rdfs:seeAlso
  - rdfs:isDefinedBy
  - rdfs:comment
  - rdfs:label

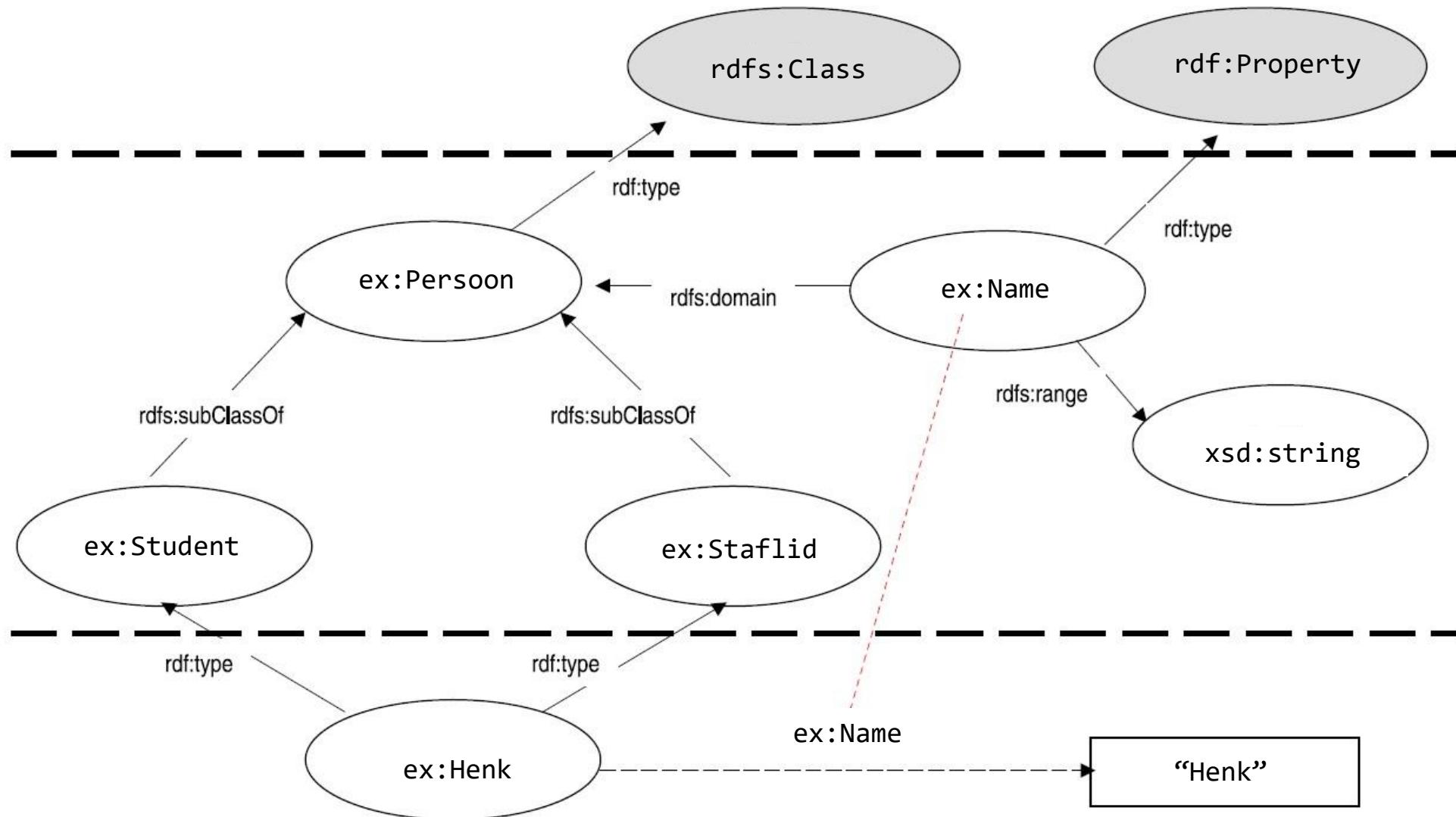
## Basis

- Classes:  
#Student rdf:type rdfs:Class
- Class hierarchie :  
#Student rdfs:subClassOf #Person
- Properties:  
#hasName rdf:type rdf:Property
- Property hierarchie:  
#hasMother rdfs:subPropertyOf #hasParent
- Properties aan classes:
  - “De property #hasName hoort alleen bij #Persoon”  
#hasName rdfs:domain #Person
  - “The type of the property #hasName is #xsd:string”  
#hasName rdfs:range xsd:string

# 'Vocabulaire'

RDFS

RDF



# SPARQL Protocol and RDF Query Language

<https://www.w3.org/TR/sparql11-query/>

## SPARQL

- W3C-standaard
- Geïmplanteerd in software
- SQL-achtig
- Querytaal voor Linked Data
- Schaalbaar (in geheugen)
- Ondersteunt federatieve queries

## Basis

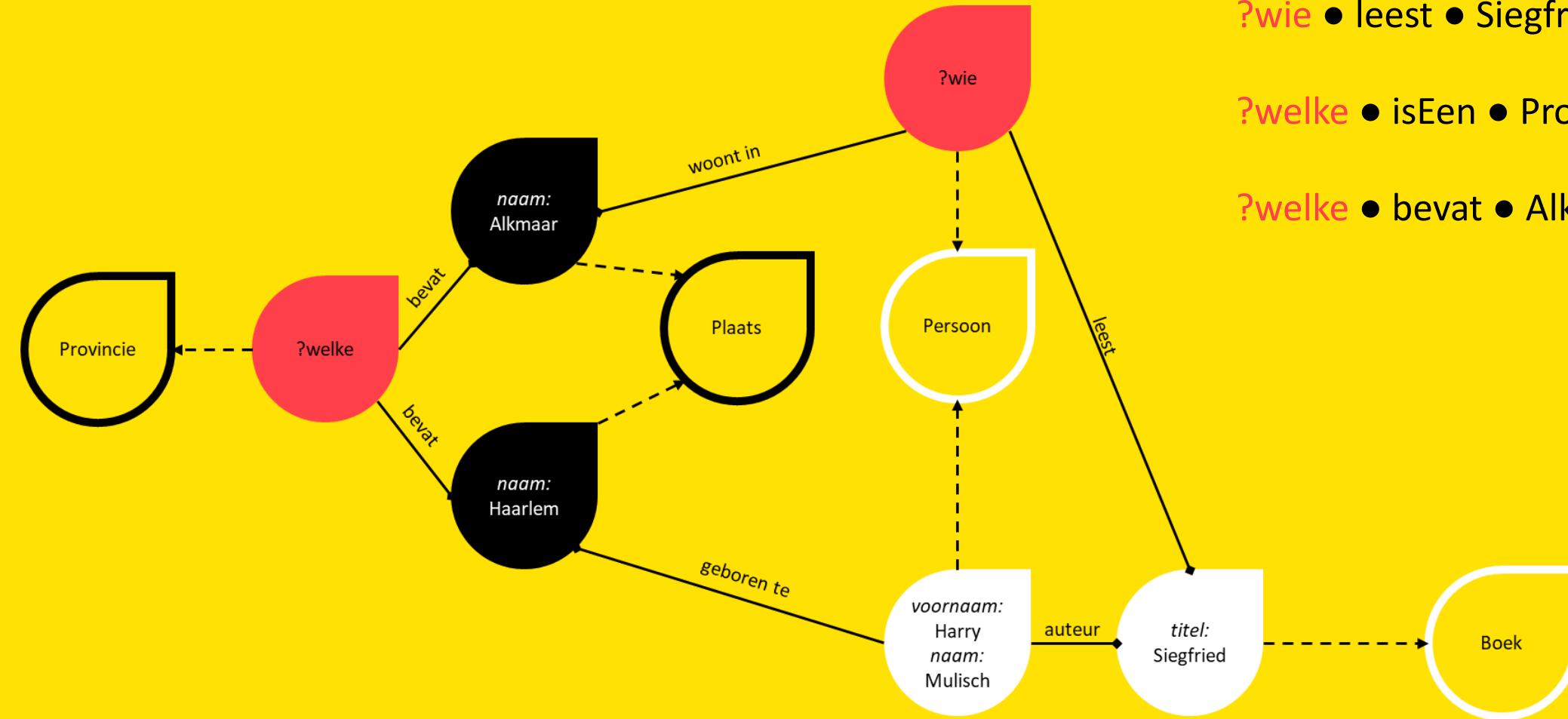
- ? voor variabelen
- SELECT voor selectie,  
WHERE voor conditie
- Alles is triples dus:  
“Triple patroon matchen”
- En variabelen gebruiken voor wat  
ingevuld moet worden

## Match alles

```
SELECT *
WHERE {
  ?subject ?predicate ?object .
}
LIMIT 10
```



# SPARQL



# SPARQL

## Selectie van ‘belangrijkste’ onderdelen

**SELECT**

- Uitleg: Gebruikt om specifieke variabelen te selecteren uit de RDF-datasets.
- Voorbeeld query:

```
SELECT ?s ?p ?o
WHERE {
  ?s ?p ?o.
}
```

**Resultaten:**

?s	?p	?o
John	hasAge	"28"
Jane	hasName	'Jane'
Bob	hasAge	"35"

**CONSTRUCT**

- Uitleg: Gebruikt om nieuwe RDF-triples te construeren gebaseerd op de resultaten van de query.
- Voorbeeld query:

```
CONSTRUCT {
  ?book ex:hasInformation ?info .
  ?info dc:title ?title ;
        dc:creator ?author .
}
WHERE {
  ?book a ex:Book ;
        dc:title ?title ;
        dc:creator ?author .
}
```

**Triples:**

```
ex:book1 ex:hasInformation ex:book1Info .
ex:book1Info dc:title "SPARQL by Example" ;
              dc:creator "John Doe" .

ex:book2 ex:hasInformation ex:book2Info .
ex:book2Info dc:title "Semantic Web Programming" ;
              dc:creator "Jane Smith" .
```

**ASK**

- Uitleg: Geeft een booleanse waarde terug (true/false) om te bepalen of de query condities voldoen.
- Voorbeeld query:

```
ASK {
  ex:book1 ex:hasInformation ex:book1Info .
}
```

**Resultaat: true**

**INSERT**

- Uitleg: Wordt gebruikt om nieuwe data toe te voegen aan de RDF store.
- Voorbeeld query:

```
INSERT DATA {
  :Alice :hasName "Alice" .
}
```

**Triples:**



# Voorbeelden query

Match alles

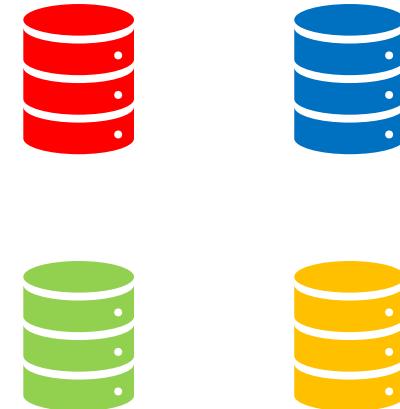
CB23 voorbeeld

Wikidata voorbeeld

# LinkedData: the Game



- 4 ‘bronnen’ die data ter beschikking stellen
- Elk een eigen server, met een eigen adres
- 2 ‘datasets’ per bron
- Alles in de vorm van (RDF) triples
- SPARQL vragen als triple patronen



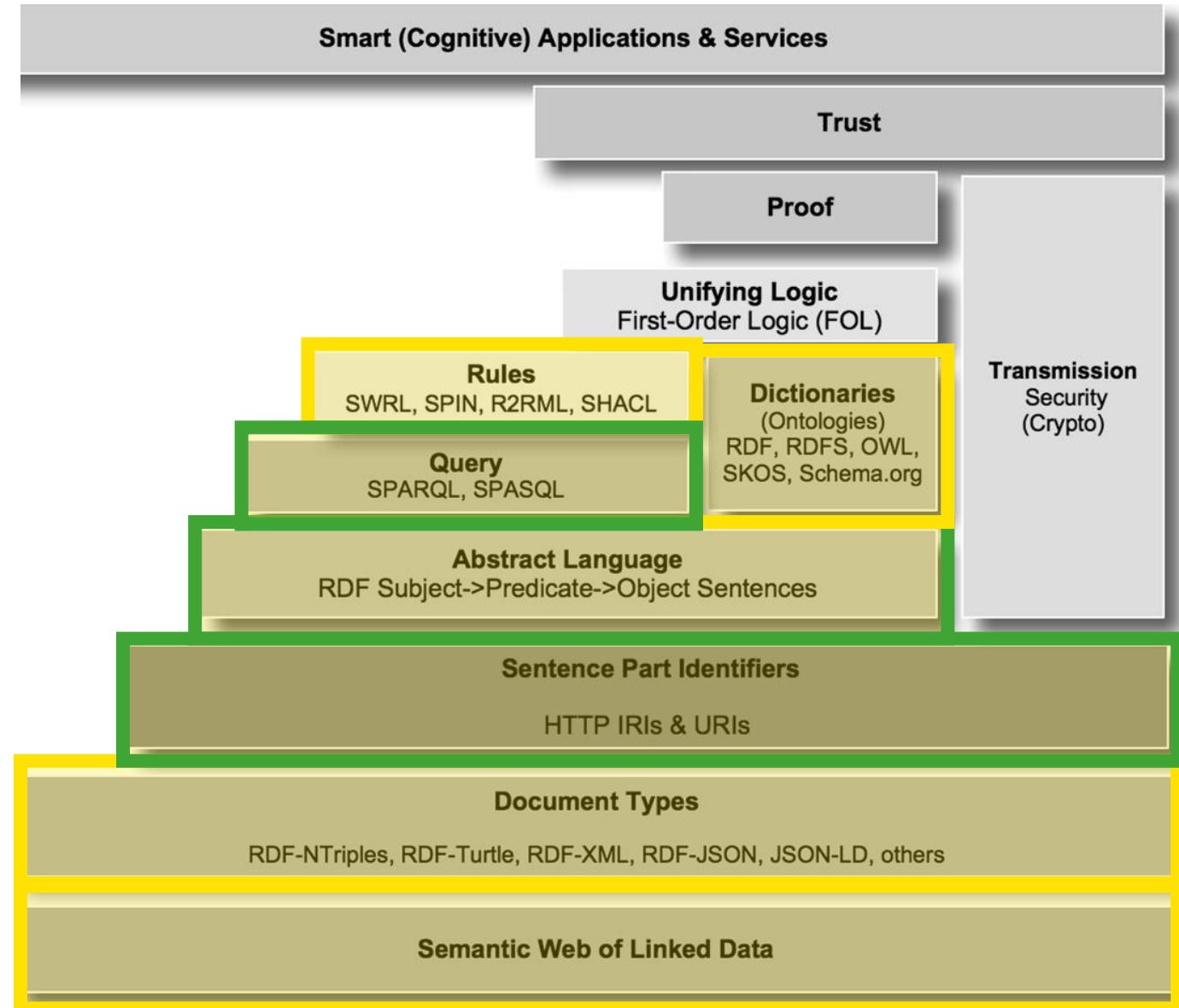
### Doel:

*Doormiddel van standaarden, gedistribueerde databronnen met onderling gelinkte data, in samenhang bevragen en zo meer kennis verzamelen als geheel*

?wie	isVanType	bag:Gebouw
?wie	isVanType	bag:Verblijfsobject
?wie	isVanType	bag:Verblijfsobject
?wie	isGemaaktVan	?wat
?wie	isVanType	bag:Verblijfsobject
?wie	isGemaaktVan	?wat
?wat	?*	?*
?waar	inProvincie	cbs:Gelderland
?welke	ligtIn	?waar
?welke	isGemaaktVan	mat:Zand
?wie	staatAan	?welke
?welke	isVanType	nwb:Straat
?welke	isGemaaktVan	?wat
?wat	bestaatUit	?*



# Semantic web stack



# Reasoning (redeneren)

## Doel

- Complexe vragen beantwoorden en verbanden leggen tussen verschillende stukjes informatie, waardoor het semantische web krachtig wordt voor geavanceerde zoekopdrachten, data-analyse en automatisering.
- Kort gezegd, reasoning maakt het mogelijk om automatisch nieuwe informatie af te leiden door logische regels toe te passen op gestructureerde gegevens.

## Inferentie

Betekent dat een computer logische regels en semantische relaties kan gebruiken om nieuwe informatie af te leiden die niet expliciet in de gegevens staat.

## Voorbeeld

In RDF staat:

- "Socrates is een mens"
- "Alle mensen sterven"

Met redeneren kan het systeem concluderen dat "*Socrates sterfelijk is*", door de regel "*Als iets een mens is en alle mensen sterven, dan is dat ding sterfelijk*".

# Web Ontology Language (OWL)

<https://www.w3.org/TR/owl2-overview/>

## Doel

- Ontworpen voor het maken van ontologieën op het web
- Definiëren van complexe gegevensstructuren
- Bieden van rijke beschrijvingen van webinhoud
- Verbeteren van gegevensintegratie en interoperabiliteit
- OWA (Open world assumption)

## Inhoud

- Klassen: categorieën van dingen
- Individuen: specifieke objecten in klassen
- Eigenschappen: relaties tussen individuen
- Axioma's: Logische (wiskundige) regels die waarheden over klassen en eigenschappen beschrijven

## Inhoud

- Restricties: Beperkingen op hoe eigenschappen worden gebruikt
- Cardinaliteit: Het aantal keer dat een eigenschap kan voorkomen (bijv. een persoon kan maximaal één echtgenoot hebben).
- Domain en range: Specificeert de klassen waaruit eigenschapswaarden moeten komen (bijv. de eigenschap "heeftLeeftijd" heeft een domein van "Persoon" en een bereik van "Getal").

Definiëren dat elke persoon die een vader heeft ook een persoon is.

- **Klassen:**

Persoon

- **Eigenschappen:**

heeftVader een eigenschap die een relatie tussen twee personen aangeeft

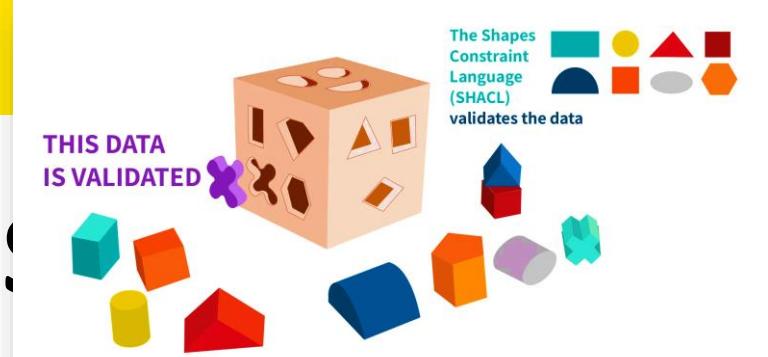
- **Axioma:**

Elk individu in de klasse Persoon moet een relatie hebben via de eigenschap heeftVader met een ander individu in de klasse Persoon.

```
ex:Persoon    rdf:type        owl:Class .  
ex:heeftVader rdf:type        owl:ObjectProperty ;  
                rdfs:domain ex:Persoon ;  
                rdfs:range  ex:Persoon .  
ex:Persoon    rdfs:subClassOf [  
                rdf:type        owl:Restriction ;  
                owl:onProperty ex:heeftVader ;  
                owl:allValuesFrom ex:Persoon  
] .
```

# Shapes Constraint Language (SHACL)

<https://www.w3.org/TR/shacl/>



## Doel

- Valideren van RDF data tegen vooraf gedefinieerde regels (shapes)
- Controleren of data voldoet aan bepaalde structurele en inhoudelijke eisen
- Consistentie: Helpt bij het handhaven van consistente data.
- Maakt geautomatiseerde data validatie mogelijk.

## Inhoud

- Shapes:  
Beschrijven de structuur van RDF data en bepalen welke eigenschappen en waarden toegestaan zijn
- Node Shapes:  
Definiëren regels voor RDF resources (bijv. "een persoon moet een naam hebben")
- Property Shapes:  
Definiëren regels voor eigenschappen van RDF resources (bijv. "een persoon moet een leeftijd hebben dat een geheel getal is".)

## Functie

- Constraints:  
Bepaalde voorwaarden waaraan data moet voldoen (bijv. datatype, waarde-bereik, vereiste aanwezigheid)
- Validation:  
Proces van het toetsen van de data aan de shapes. Geeft fouten en waarschuwingen als data niet voldoet.



Een RDF dataset valideren waarbij elke persoon een naam en een leeftijd moet hebben. De naam moet een string zijn en de leeftijd moet een geheel getal zijn tussen 0 en 150.

```
ex:PersonShape    rdf:type      sh:NodeShape ;
                   sh:targetClass ex:Person ;
                   sh:property [
                     rdf:type      sh:PropertyShape ;
                     sh:path       ex:name ;
                     sh:datatype   xsd:string ;
                     sh:minCount   1 ;
                   ] ;
                   sh:property [
                     rdf:type      sh:Propertyshape
                     sh:path       ex:age ;
                     sh:datatype   xsd:integer ;
                     sh:minCount   1 ;
                     sh:maxCount   1 ;
                     sh:minInclusive 0 ;
                     sh:maxInclusive 150 ;
                   ] .
```

# Vocabulaires



- Simple Knowledge Organization System
- Model voor het uitdrukken van basis structuur en content
- Gebruikt voor thesauri, taxonomieën en vocabulaires
- Gebruikt RDF
- Gestandaardiseerd door W3C

skos:Concept  
skos:broader  
skos:narrower  
skos:prefLabel  
...  
...



- Vocabulaire voor gestructureerde data
- Gecreëerd door Google, Microsoft, Yahoo, en Yandex
- Beschrijft entiteiten en relaties in de context webpagina's
- Verbeterd zoekresultaten en zichtbaarheid
- Veel gebruikt voor zoekmachines

schema:Person  
schema:name  
schema:knows  
schema:description...



- Dublin Core Metadata Terms
- Vocabulaire voor metadata
- Beschrijft resources (bijv. documenten, datasets)
- Bevat termen zoals titel, auteur, datum
- Gestandaardiseerd door DCMI (Dublin Core Metadata Initiative)
- Breed toegepast voor documentbeheer en digitale bibliotheken

dct:Event  
dct:source  
dct:created  
dct:hasPart  
...  
...

# Syntax / formaten

Turtle (.ttl)



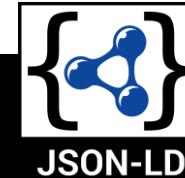
```
@prefix schema: <http://schema.org/> .  
  
<http://example.org#henk> a schema:Person ;  
    schema:hasName "Henk Smit" .
```

N-Triple (.nt)



```
<http://example.org#henk> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://schema.org/Person> .  
<http://example.org#henk> <http://schema.org/hasName> "Henk Smit" .
```

JSON-LD (.json-ld)



```
[  
 {  
     "@id": "http://example.org#henk",  
     "@type": [  
         "http://schema.org/Person"  
     ],  
     "http://schema.org/hasName": [  
         {  
             "@value": "Henk Smit"  
         }  
     ]  
 }  
]
```

```
@prefix schema: <https://schema.org/> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix me: <http://rik.ik/> .  
@prefix nen2660: <https://w3id.org/nen2660/def#> .  
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .  
  
me:RO a schema:Person;  
    rdfs:label "Rik"@nl, "Rik"@en;  
    schema:hasOccupation me:Tr;  
    schema:worksFor me:CROW;  
    schema:owns me:zp243x .  
  
me:Tr a schema:Occupation;  
    rdfs:label "Trainer"@nl, "Trainer"@en;  
    schema:skills "LinkedData, Semantic Web" .  
  
me:CROW a schema:Organization;  
    rdfs:label "Kennisplatform CROW"@nl, "Knowledge Platform CROW"@en;  
    schema:foundingDate "1987-04-21"^^xsd:date .  
  
me:zp243x a schema:Car, nen2660:PhysicalObject;  
    rdfs:label "Mijn Mazda 6"@nl, "My Mazda 6"@en;  
    nen2660:hasPart me:zp243_motor .  
  
me:zp243_motor a nen2660:PhysicalObject;  
    rdfs:label "2.0 Sky-Active"@en .
```

<https://issemantic.net/rdf-converter>

# LinkedData: hoe maak je het

Data beschikbaar stellen als RDF



# Gebouwde omgeving en LinkedData

## Delen van ontologie

Ofwel het beschikbaar stellen van het informatiemodel

```
imbor:Bierleiding      a  
imbor:Bierleiding      skos:definition  
imbor:Bierleiding      imbor:Attribuut  
imbor:Bierleiding      imbor:Attribuut  
...  
imbor:Bierleiding      rdfs:seeAlso  
imbor:Bierleiding      rdfs:subClassOf  
imbor:Bierleiding      rdfs:subClassOf  
imbor:Bierleiding      rdfs:subClassOf
```

```
gsws:TransportLeiding gsws:Attribuut  
...  
gemeenteX:LeidingGrolsch-ZwarteCross  
gemeenteX:LeidingGrolsch-ZwarteCross  
gemeenteX:LeidingGrolsch-ZwarteCross  
gemeenteX:LeidingGrolsch-ZwarteCross
```

## Linken van ontologieën

Ofwel het verbinden van informatiemodellen via eenzelfde syntax

```
rdf:Class  
“Leiding voor het transport van bier.”  
imbor:wandruwheid  
imbor:voegmateriaal
```

```
imgeo:bor-type-bierleiding  
nen3610:Leiding  
nen2660:FysiekObject  
gsws:TransportLeiding  
gsws:diameterLeiding
```

```
a  
gsws:diameterLeiding  
imbor:wandruwheid  
imbor:voegmateriaal
```

## Delen van gegevens

Ofwel het delen van de dataset op basis van het informatiemodel

```
imbor:Bierleiding  
“15 cm”  
“100,9”  
“Rubbering”
```

# NEN2660-2

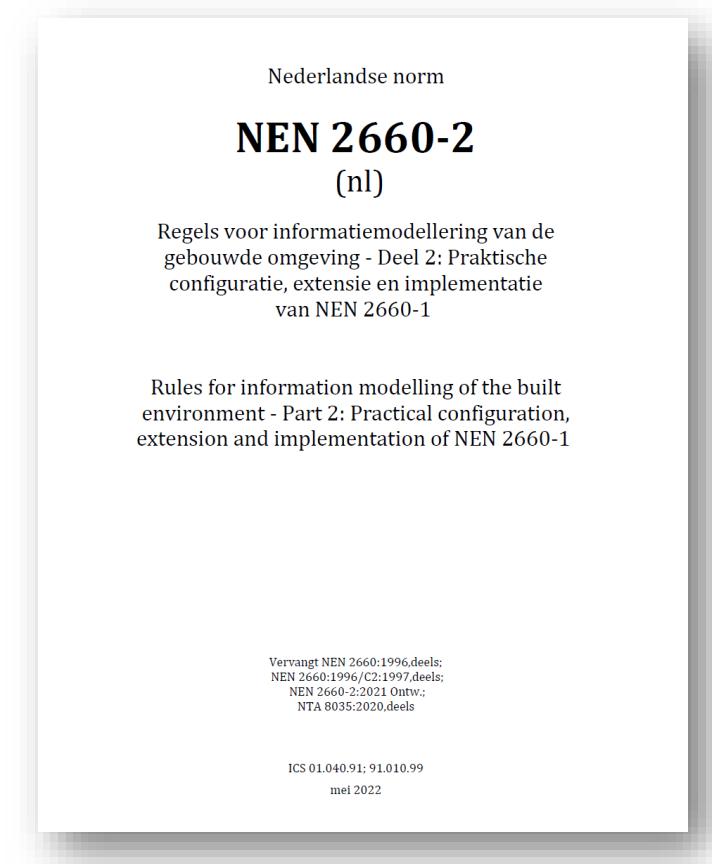
Complexere interdisciplinaire projecten

Bijvoorbeeld: een straat gaat open

- veel stakeholders, dus veel input
- meerdere disciplines, dus veel data delen
- veel eisen, dus machine ondersteuning

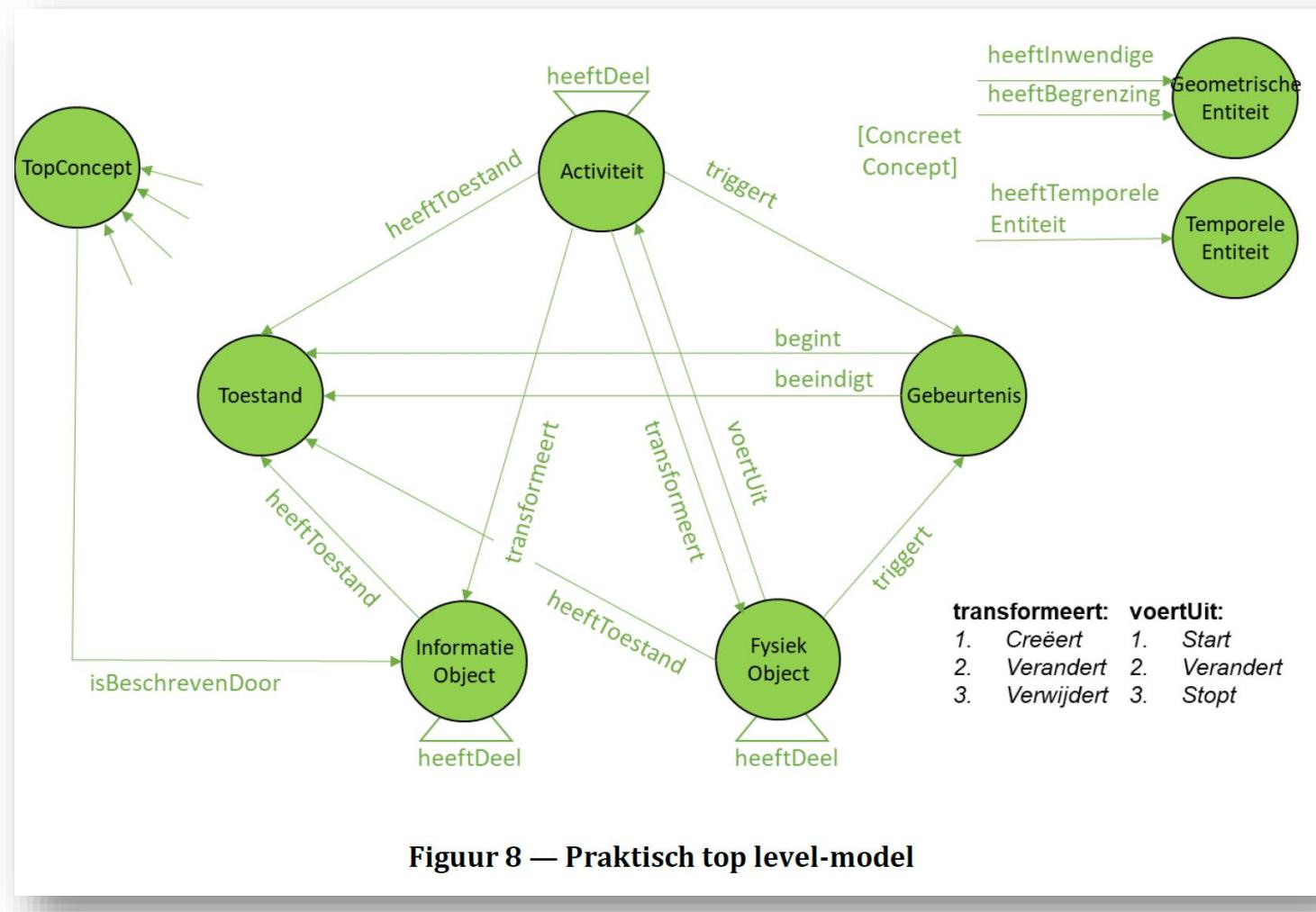
## NEN2660-2

1. Een praktisch toplevelmodel waarin genoeg semantiek aangegeven wordt
2. Een taalbinding (en daarmee de keuze voor) de LinkedData W3C standaarden



Tabel 2 — Metamodelconfiguratie NEN 2660-2

<b>Metaverzameling</b>		
1. Informatiemodel		
2. Informatieverzameling		
3. Groep		
<b>Metaconcept</b>		
4. Concept		
5. Individu		
6. Waardetype		
7. Waarde		
8. Attribuut		
8.1 Annotatie		
8.2 Kwaliteit		
8.3 Kwantiteit		
9. Relatie		
<b>Metarelatie</b>		<b>Van</b>
10. Groepering 'heeftLid'	1 tot 3 (of)	3 tot 9 (of)
11. Classificatie (inverse: Instantiatie) 'isVanType'	4 of 5 7	4 6
12. Generalisatie (inverse: Specialisatie) 'isSubtypeVan'	4 6 8 9	4 6 8 9
13. Beperking	(complexe n-aire relatie)	
14. Afleiding	(complexe n-aire relatie)	



Figuur 8 — Praktisch top level-model

NEN 2660-2:2022

Metaverzameling -concept/-relatie	SKOS-taalconstructie
9. Relatie	skos:Concept
10. Groepering	skos:inScheme, skos:member
11. Classificatie	skos:broad (alleen voor enumeratietype instanties)
12. Generalisatie	skos:broad
13. Beperking	n.v.t.
14. Afleiding	n.v.t.

**OPMERKING 1** Individuen worden niet gemodelleerd in SKOS. Daarom zijn rdf:type-relaties meestal niet relevant. Er is één uitzondering: de instanties van de abstracte 'type klassen' die worden gebruikt voor enumeratietypen en objecttypen. Hier worden de instanties ook SKOS-concepten gerelateerd via skos:broad aan het relevante enumeratiotype.

Tabel 7 — Taalbinding voor RDFS, (RDFS+)OWL en (RDFS+)SHACL

Metaverzameling -concept/-relatie	RDFS	RDFS+OWL	RDFS+SHACL
1. Informatiemodel		owl:Ontology	
2. Informatieverzameling		owl:Ontology	
3. Groep		rdfs:Container	
4. Concept	rdfs:Class	owl:Class	rdfs:Class
5. Individu	rdfs:Resource (impliciet)	owl:NamedIndividual of anoniem individu (beide impliciet)	rdfs:Resource (impliciet)
6. Waardetype	rdfs:Datatype of rdfs:Class + via rdf:type-gerelateerde referentie-individuen van Typen	rdfs:Datatype of owl:Class + via rdf:type-gerelateerde referentie-individuen van Typen incl. owl:oneOf voor aanduiding <u>vaste</u> typen	rdfs:Datatype of rdfs:Class + via rdf:type-gerelateerde referentie-individuen van Typen incl. sh:in voor aanduiding <u>vaste</u> typen
7. Waarde	plain of typed literal (impliciet)	plain of typed literal (impliciet)	plain of typed literal (impliciet)

NEN 2660-2:2022

Tabel 8 — URI-strategie/syntax

uri =	'http', '['s'], '://', internet domein, '/', [pad], '/', referentie
referentie =	
	'term', ('/'   '#'), term <sup>a</sup>
	def, ('/'   '#'), (conceptnaam   attribuutnaam   relatienaam   referentie-individu-naam) <sup>b</sup>
	'id', ('/'   '#'), individu-naam <sup>c</sup>
	'doc', ('/'   '#'), documentnaam, [documentextensie], [( '/'   '#'), fragmentidentificatie] <sup>d</sup>
<b>Metasymbolen</b>	
=	wordt gedefinieerd door (een 'productieregel')
,	symboolscheider
'...'	voorgedefinieerd symbool ('terminal symbol')
...	verplicht symbool
[...]	optioneel symbool
(...)	groepering
	logische 'exclusieve OR'
<sup>a</sup> Om een term voor een concept, attribuut of relatie aan te geven.	
<sup>b</sup> Om een concept, attribuut, relatie of referentie-individu aan te geven.	
<sup>c</sup> Om een individu aan te geven (lees voor 'id' hier 'individu' in plaats van 'identifier' (want termen, semantische definities en documenten hebben ook een identifier).	
<sup>d</sup> Om een document met informatie over een concept, attribuut, relatie of (referentie-)individu aan te geven (optioneel met fragmentidentificatie binnen het document).	

VOORBEELD 1 http://www.interlink.eu/roadotl/def/Road

VOORBEELD 2 http://www.nwb-roadnetwork.nl/id#A16

VOORBEELD 3 http://www.interlink.eu/doc#RoadDefinition.docx

VOORBEELD 4 https://w3id.org/nen2660/def#PhysicalObject

(https://w3id.org/nen2660/def# is de 'name space URI')

VOORBEELD 5 https://w3id.org/nen2660/shacl/def (voorbeeld van een 'graph URI')

**OPMERKING 1** De keuze tussen '/' of '#' hangt onder andere af van de verwachte omvang van de informatiemodellen/informatieverzamelingen. Indien deze relatief groot zijn, gebruik dan '/'. Na een '#' is er geen HTTP server control meer.

58

## 8.5 Annotatiebinding

De taalbinding voor de annotaties (zie hoofdstuk 5) wordt gegeven door beschikbare RDFS- en SKOS-annotaties. Alleen 'afkorting' is in deze vocabulaires niet beschikbaar en wordt gemapt naar een zelf gedefinieerde nen2660:afkorting, wel als subtype van de bestaande skos:altLabel.

Een overzicht van de geselecteerde annotatietypen die in dit document worden gespecificeerd, wordt gegeven in tabel 9.

Tabel 9 — Taalbinding voor annotaties

Annotatie	Taalbinding
Primaire label	skos:prefLabel
Synoniem	skos:altLabel
Code	skos:notation
Afkorting	nen2660:abbreviation (rdfs:subPropertyOf skos:altLabel)
Interne definitie	skos:definition (max één per taal; textueel)
Externe definitie	rdfs:isDefinedBy (niet per se RDF-gebaseerd)
Externe beschrijving	rdfs:seeAlso (niet per se RDF-gebaseerd; zwakker dan externe definitie; gebruikt voor link naar SKOS-schema)
Scope	skos:scopeNote
Commentaar	rdfs:comment
Opmerking door auteur	skos:editorialNote
Voorbeeld toepassing	skos:example

Uiteraard kunnen andere annotaties worden toegevoegd, zoals Dublin Core's 'dc:source'.

60

# Tools

*! Dit is een onvolledig overzicht van de tools die ik wel eens gebruik op een schaal die ik verzonden heb*

## Triple store / LDP

*Jena Fuseki      Laces HUB  
RDFox               GraphDB  
Virtuoso            TriplyDB  
                      YaSGUI*

## Convertor

*RML  
SPARQL Anything  
isSemantic.net  
SKOS Play !  
LD Wizard*

*Excel*

*Laces Library Manager*

*Notepad++  
Visual Studio Code*

**Text triple editor**

*VocBench  
Topbraid Composer  
(Web)Protogé*

**Ontology editor**

# Hoe gebruik je het

## Oefeningen



# Opzet van de oefeningen

- Ontology editor: Excel (o.i.d.)
- Convertor: SKOS Play ! (online)
- Turtle bestand in Text Editor
- Upload naar LDP CROW
- YasGUI (online)
- SPARQL!

# Oefening: Data maken

[t.ly/QOtPk](https://t.ly/QOtPk)

<https://github.com/nl-digigo/training-linkeddata/tree/main/Oefeningen>

**! Hoofdletter gevoelig**

*Documentatie:*

<https://xls2rdf.sparna.fr/rest/doc.html#generating-plain-rdf-not-skos>

- **Klik op excel2rdf - TEMPLATE.xlsx**
- **Klik op View raw en download**
- **Open en verander de rode tekst**
- **Vul de rijen onder groene balk in**  
*(en vul eventueel de groen balk aan)*
- **Gebruik schema.org en [w3id.org/nen2660](https://w3id.org/nen2660)**
- **Sla de file op met een herkenbare naam**  
*(bijv. excel2rdf – Rik.xlsx)*

# Oefening: RDF maken

[t.ly/DMgYk](https://t.ly/DMgYk)

<https://skos-play.sparna.fr/play/convert>

! Hoofdletter gevoelig

! Eerst Excel opslaan en afsluiten voor uploaden

- **Vink aan In a local file on my computer**
- **Klik op Select en zoek de Excel sheet**
- **Verander default language naar nl**
- **Selecteer bij Select the resulting RDF serialization het Turtle format**
- **Klik op Convert en download (Opslaan als \*.ttl)**

# Oefening: Triples bekijken en aanpassen

[t.ly/8cBYH](https://t.ly/8cBYH)

<https://felixlohmeier.github.io/turtle-web-editor/>

**! Hoofdletter gevoelig**

- Open tt1 in kladblok en kopieer naar Turtle Web Editor
- (of gebruik eigen text editor)
- Bekijk en verander je triples, voeg i.i.g. toe:
  - een schema:alternateName met je bijnaam
  - een schema:description met een mop/grap in het Nederlands
- Klik op Validate! (en eventueel Download)

# Data online zetten

t.ly/\_ink8

<https://filebin.net/z12djtadyhyykwc9>

**! Hoofdletter gevoelig**

- **Klik op More en dan op Upload more files**
- **Voeg je ttl en xlsx file toe**
- **(xlsx optioneel voor referentie)**
- **Geef je trainer even ...**

# Oefening: SPARQL

<https://yasgui.triply.cc/>

- Verander de SPARQL-Endpoint  
<https://dbpedia.org/sparql>  
in de blauwe link hieronder en druk op  
**ENTER**

<https://api.datasets.crow.nl/datasets/Rik/TLD/sparql>

- **Selecteer alle Fysieke Objecten EN hun onderdelen**
- **Laat alles zien wat er over jou bekend is**
- **Selecteer iedereen die een auto heeft**

# Eindbaas Oefening: SPARQL

Hoeveel Duitstalige boeken bestaan er in  
de Koninklijke Bibliotheek Nederland die  
exact 300 pagina's hebben. En wie  
hebben die geschreven?

<http://data.bibliotheken.nl/sparql>



# “Oefening”: Gebruik ChatGPT voor je SPARQL

<https://chatgpt.com/>

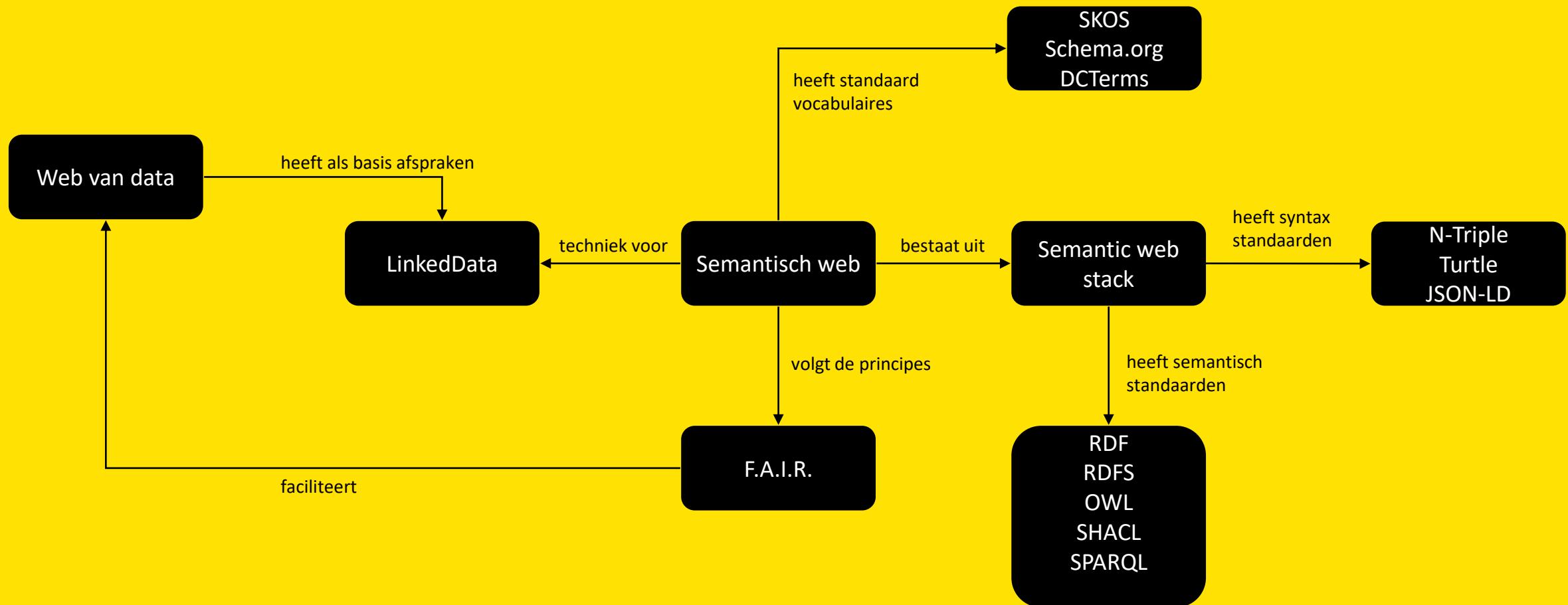
- ChatGPT (of een ander LLM) is behoorlijk goed in SPARQL query's maken en verbeteren

I want to count how many schema:Person there are in this query:

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

```
SELECT *
WHERE {
    ?sub ?pred schema:Person .
}
LIMIT 10
```

# Model van de training



# Mentions

## API en SPARQL

REST-API ≠ SPARQL-Endpoint

... bij een SPARQL-Endpoint bepaalt de vraagsteller, bij een REST-API de antwoordgever

## Toekomstmuziek

Semantic web & LinkedData ≠ Toekomstmuziek

...het bestaat sinds de 00's en wordt breed toegepast

## Domeinkennis

Het aansluiten van informatiebronnen vergt ook **domeinkennis**.

Anders maak je keuzes die je gebruikers op het verkeerde been kunnen zetten.

# Linked Data ≠ (Linked) Open Data

## Linked Data

Methode om data te publiceren op het web op een zodanige manier dat deze kan worden verbonden en gecombineerd met andere datasets. Het maakt gebruik van standaard webtechnologieën zoals HTTP en URIs, samen met RDF om data op een gestructureerde en semantische wijze te verbinden.

*Data technisch verbonden op het web.*

## Open Data

Data die vrij beschikbaar is voor iedereen om te gebruiken, te hergebruiken en te verspreiden zonder enige beperking. Dit type data is vaak openbaar, niet gepatenteerd, en wordt verstrekken onder een open licentie die weinig tot geen beperkingen oplegt.

*Vrij toegankelijke data zonder gebruiksbeperkingen.*

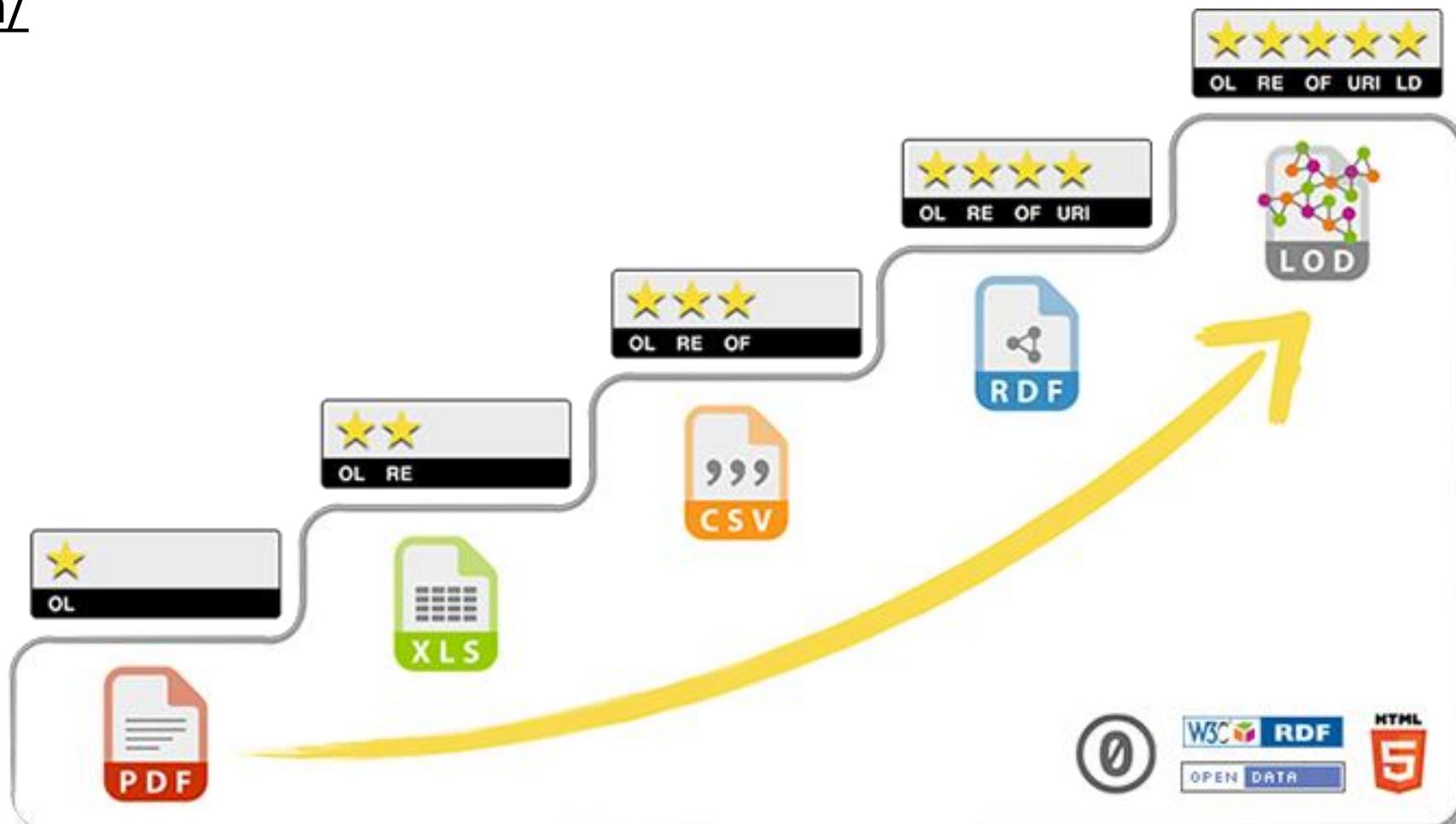
## Linked Open Data

Een combinatie van de concepten van Linked Data en Open Data. Het verwijst naar open data die gepubliceerd is volgens de principes van Linked Data. Dit betekent dat de data vrij toegankelijk is en verbonden kan worden met andere datasets, waardoor een rijke, verbonden datastructuur ontstaat.

*Vrij toegankelijke data die technisch verbonden is met andere data.*

# 5 – sterren open data

<https://5stardata.info/en/>



# Links

- <https://issemantic.net/rdf-converter>  
*Handig online RDF convertor*
- <https://simplea.com/Articles/semantic-web>  
*Leuk en informatief artikel over Semantic web*
- <https://csiro-enviro-informatics.github.io/info-engineering/standards.html>  
*Handig overzicht van alle Semantic web standaarden*
- <https://www.ontotext.com/knowledgehub/fundamentals/what-are-ontologies/>  
*Informatieve blog over ontologieën*
- <https://w3id.org/nen2660/>
- *Alle ‘LinkedData’ informatie van de NEN2660-2*

# Bedankt voor de aandacht en de medewerking!



<https://rix.fyi/>