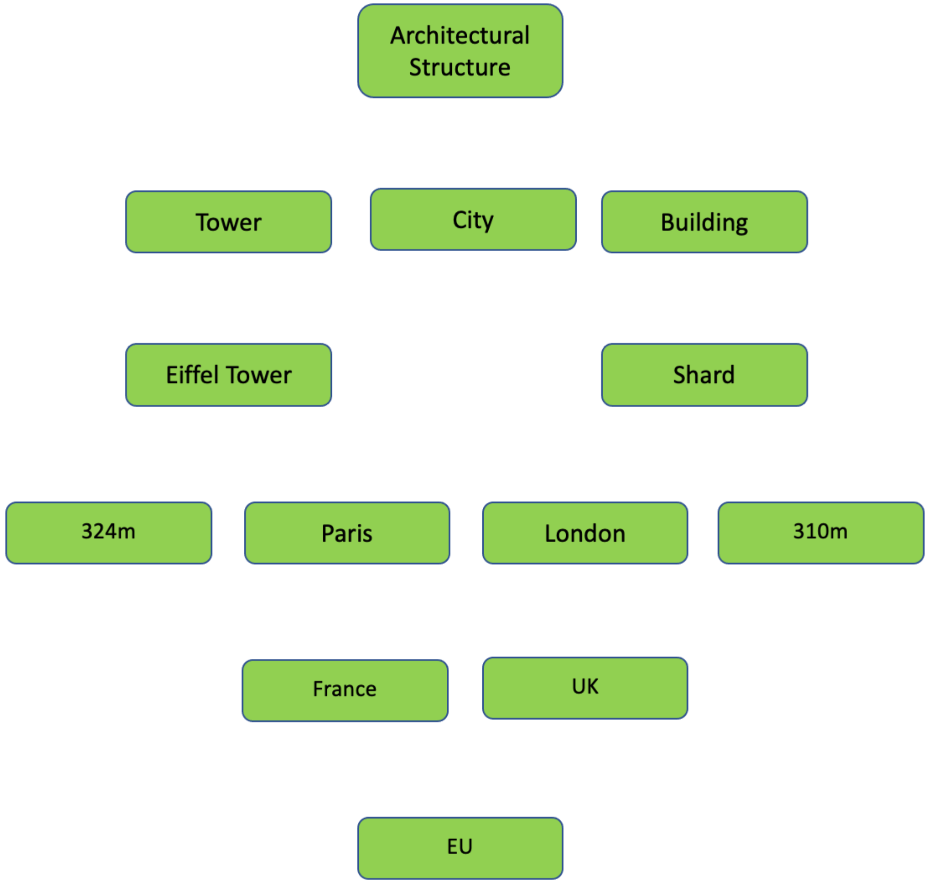
Reasoning over Knowledge Graphs: Motivation, Theory and Practice

Ian Horrocks

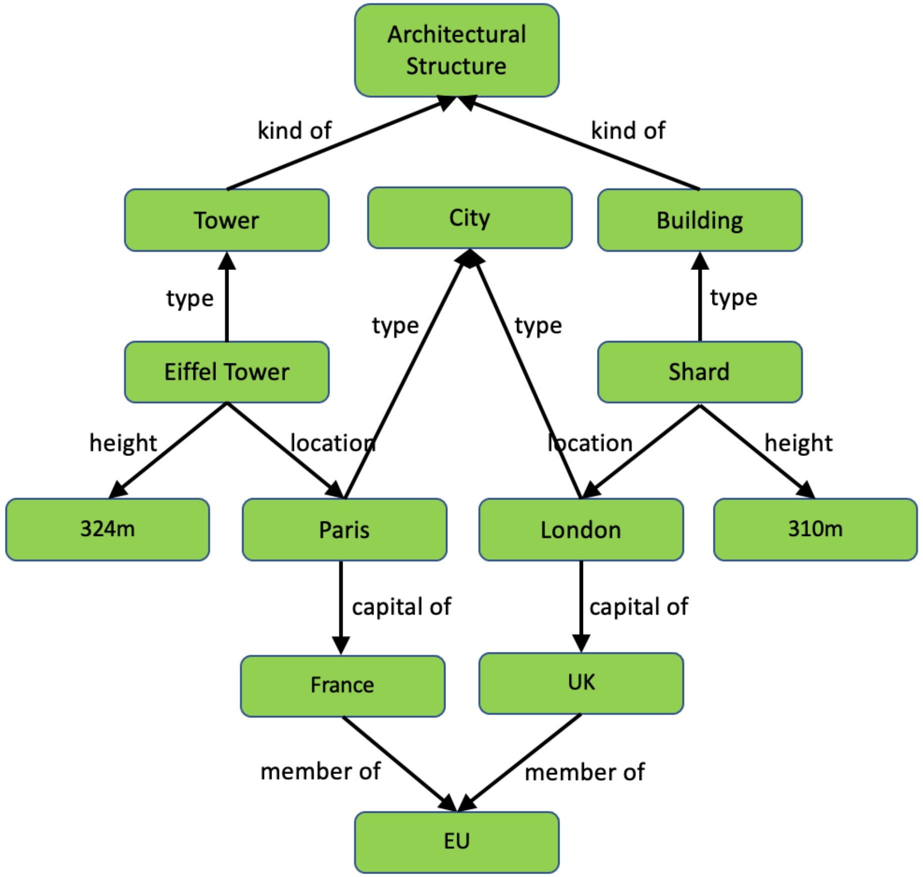
UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

Introduction to Knowledge Graphs

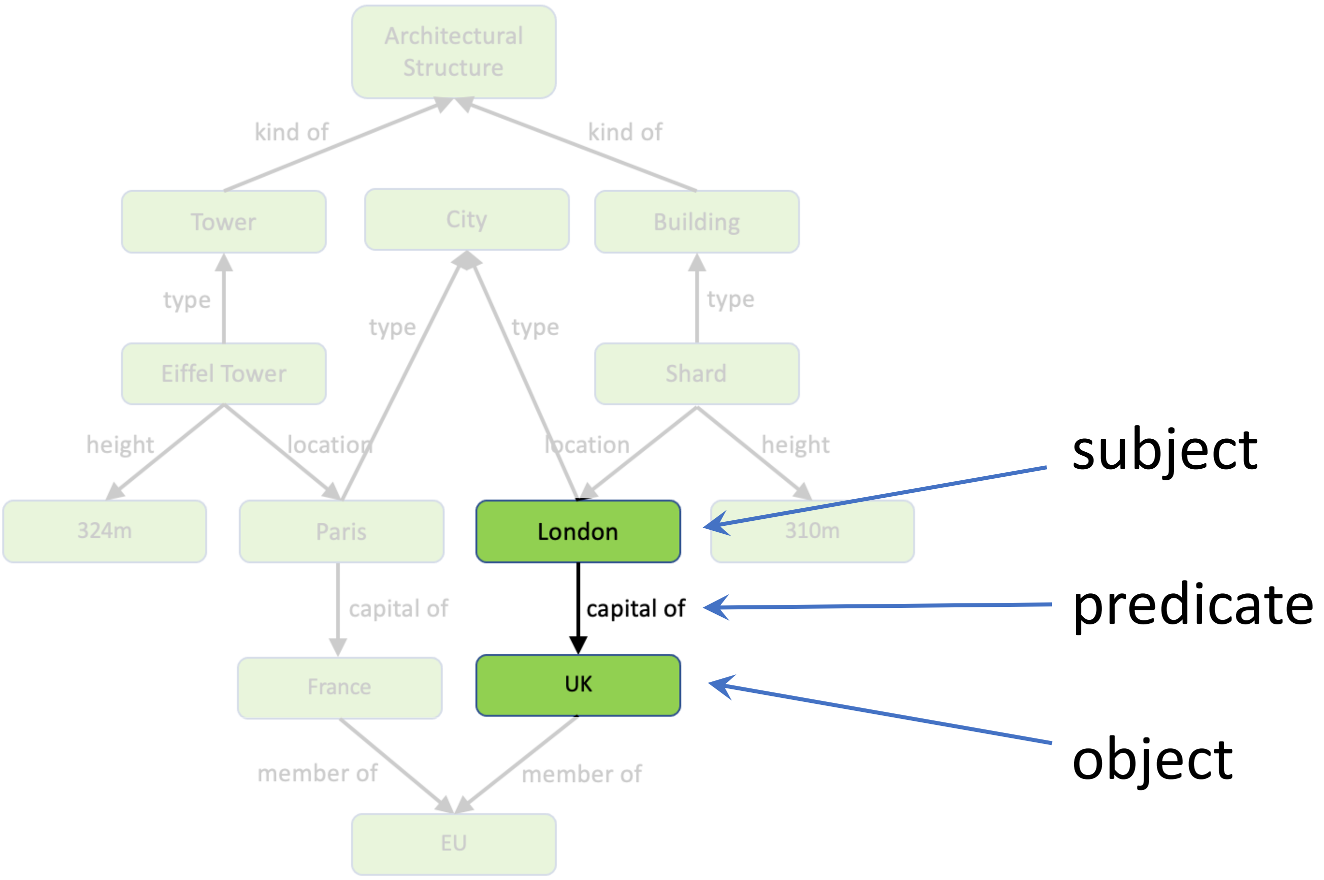
Anatomy of a Knowledge Graph



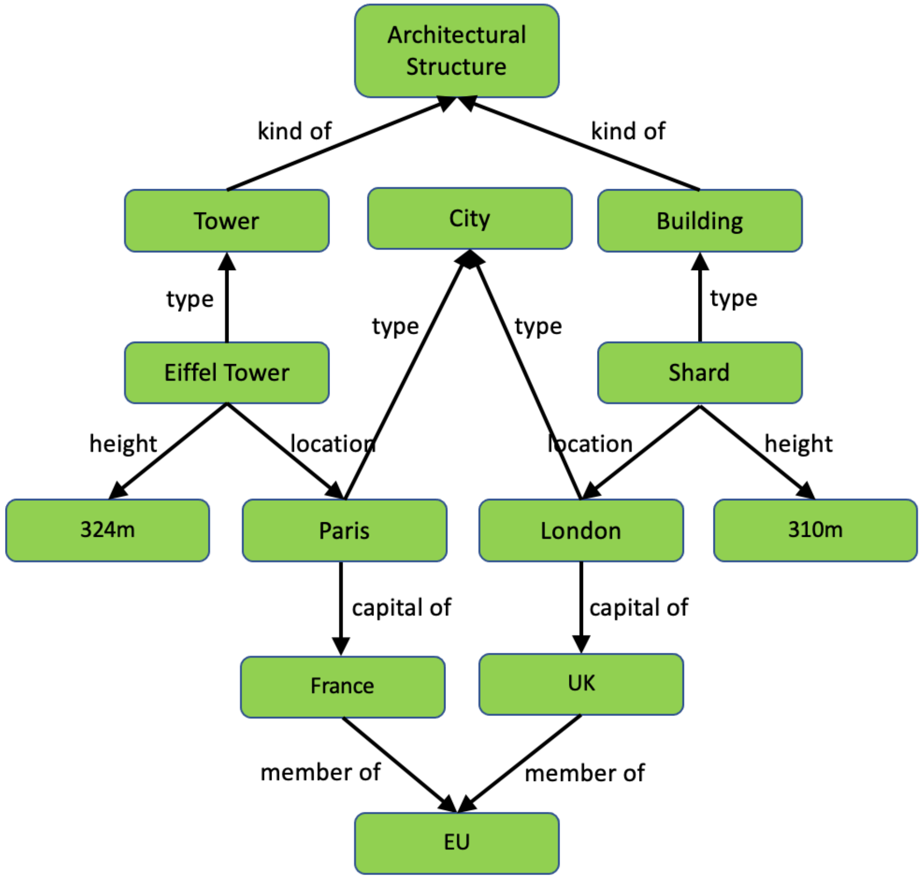
Anatomy of a Knowledge Graph



Anatomy of a Knowledge Graph



Anatomy of a Knowledge Graph



$\sim$

Architectural Structure		
Name	Height	Location
Eiffel Tower	324	Paris
Shard	310	London

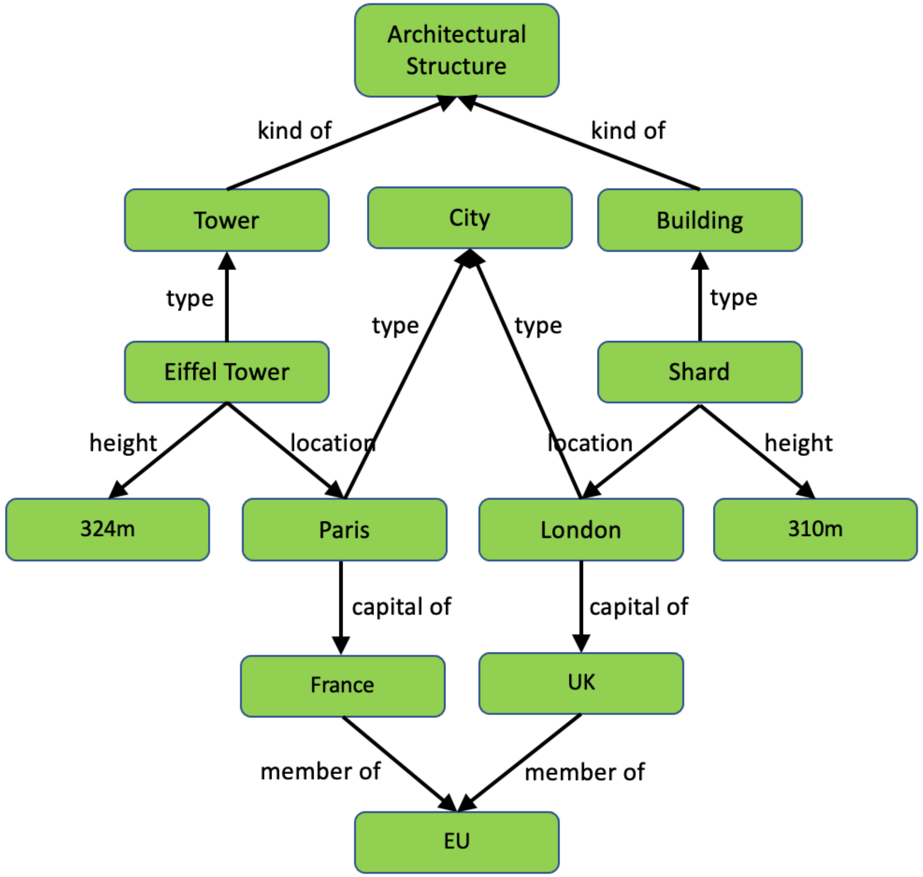
Tower
Name
Eiffel Tower

Building
Name
Shard

City	
Name	Capital Of
Paris	France
London	UK

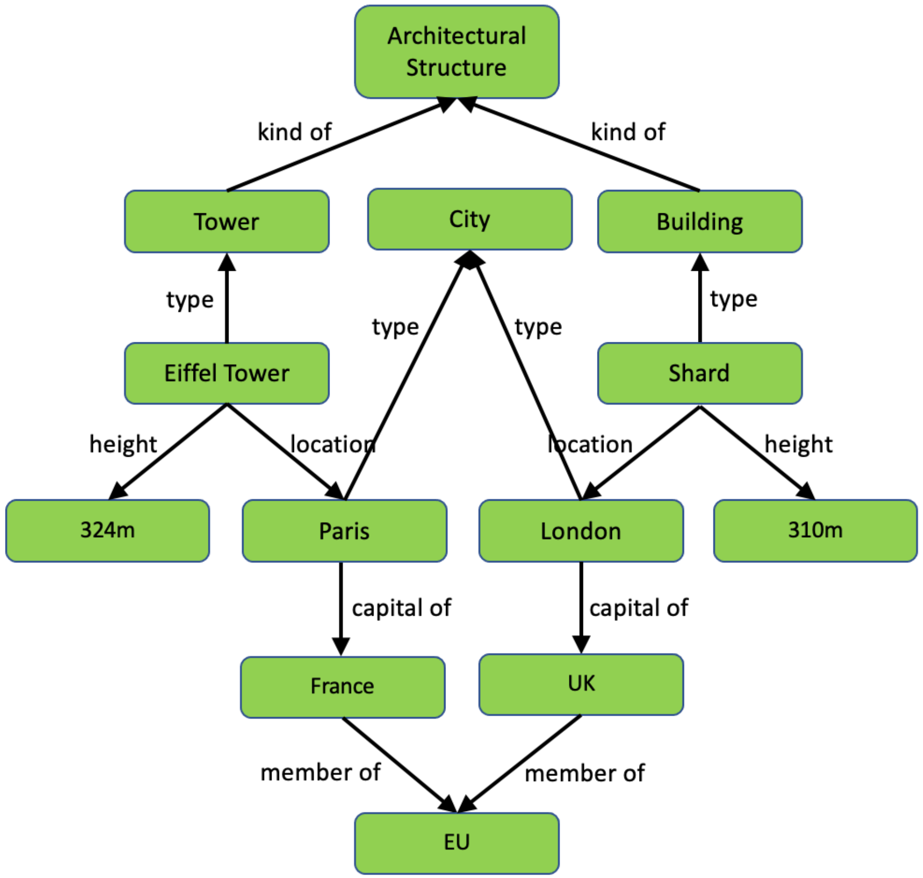
Member	
Country	Union
France	EU
UK	EU

Anatomy of a Knowledge Graph



✓ Intuitive (e.g., no “foreign keys”)

Anatomy of a Knowledge Graph



Architectural Structure		
Name	Height	Location
Eiffel Tower	324	Paris
Shard	310	London

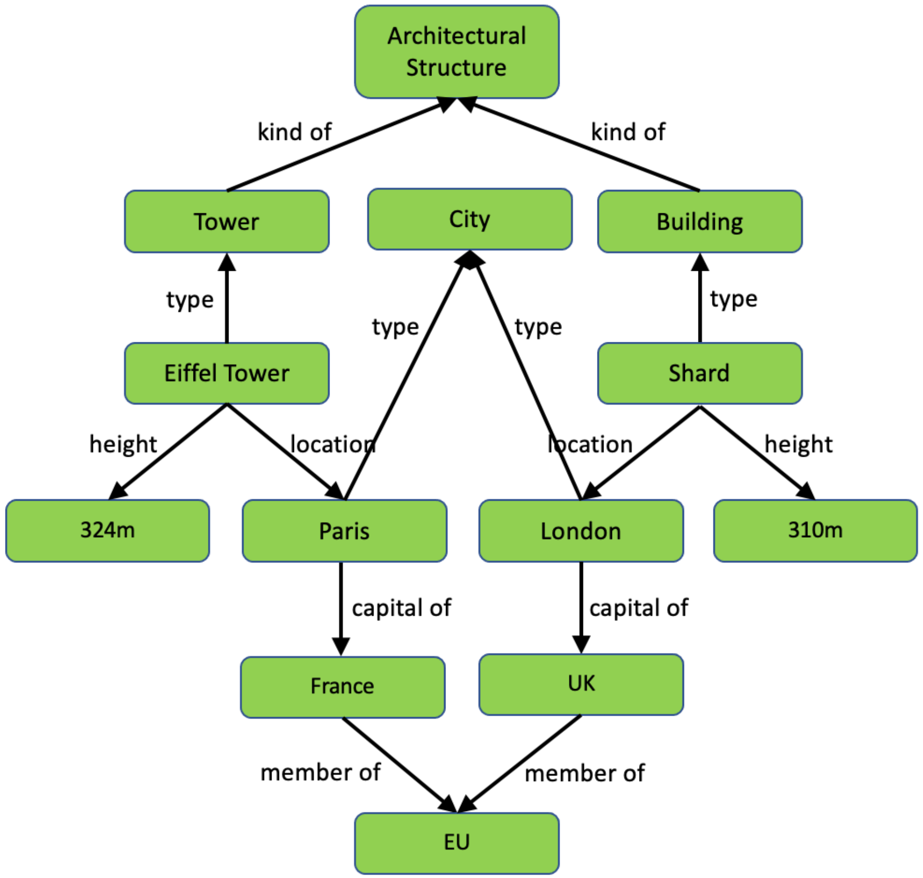
Tower
Name
Eiffel Tower

Building
Name
Shard

City	
Name	Capital Of
Paris	France
London	UK

Member	
Country	Union
France	EU
UK	EU

Anatomy of a Knowledge Graph



Architectural Structure		
Name	Height	Location
Eiffel Tower	324	Paris
Shard	310	London

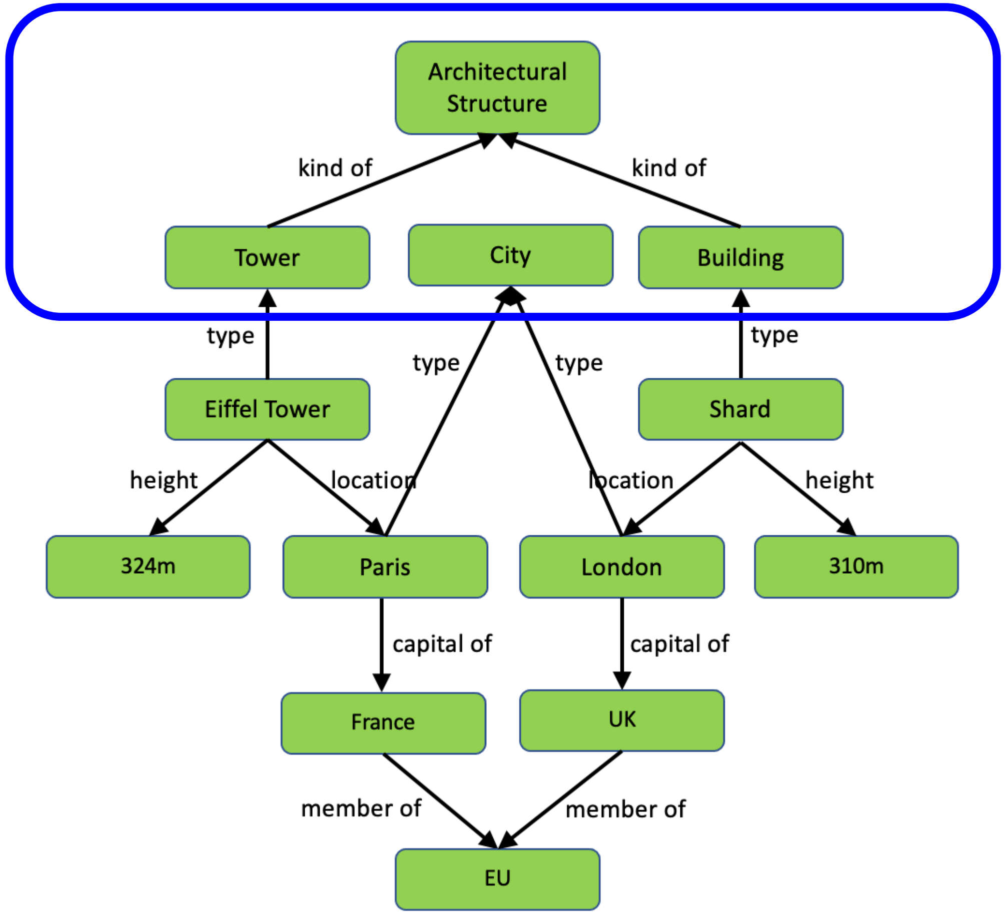
Tower
Name
Eiffel Tower

Building
Name
Shard

City	
Name	Capital Of
Paris	France
London	UK

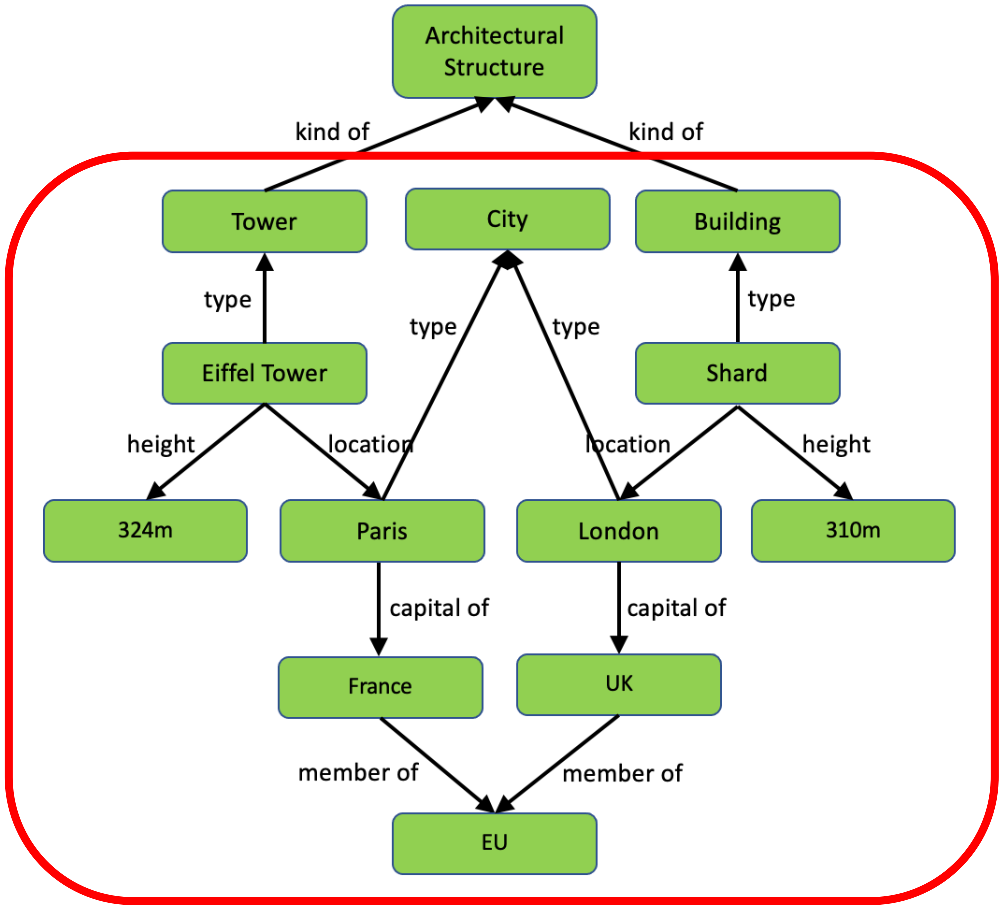
Member	
Country	Union
France	EU
UK	EU

Anatomy of a Knowledge Graph



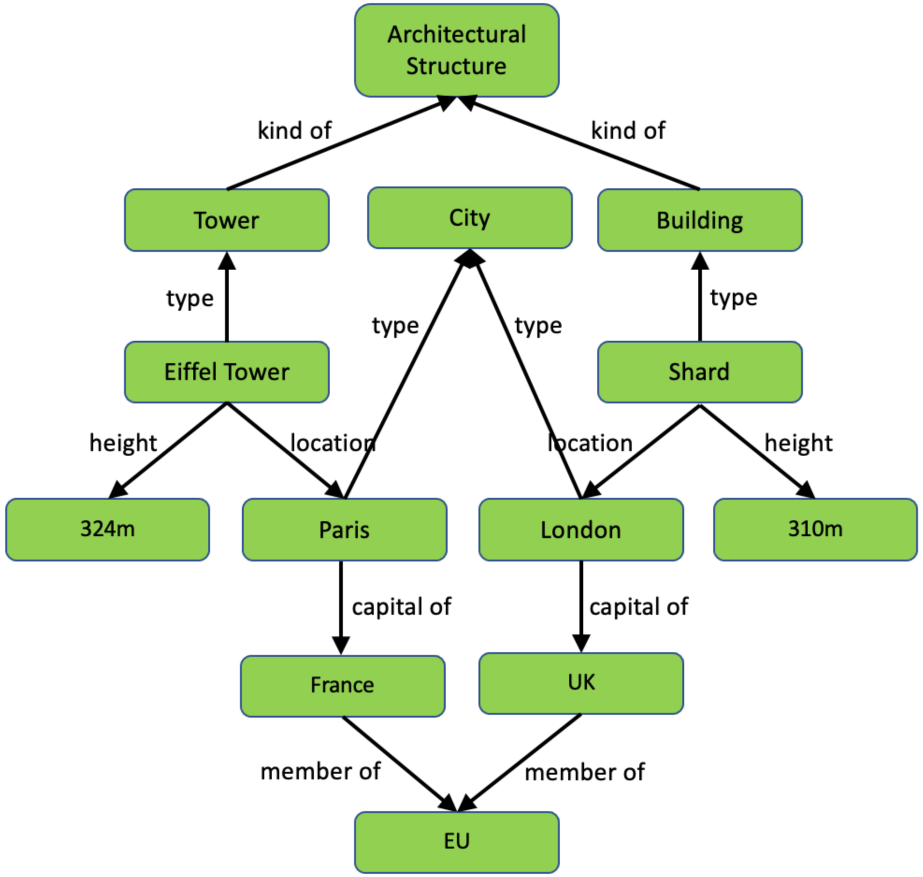
- ✓ Intuitive (e.g., no “foreign keys”)
- ✓ **Data** + **schema (ontology)**

Anatomy of a Knowledge Graph



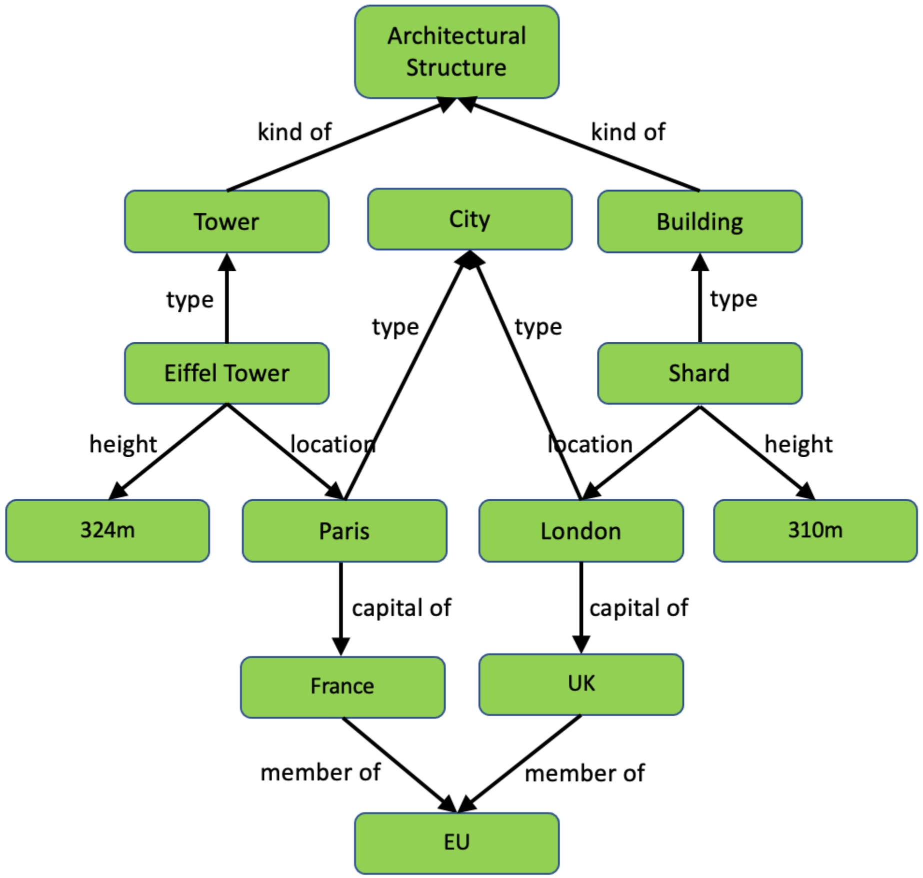
- ✓ Intuitive (e.g., no “foreign keys”)
- ✓ **Data** + **schema (ontology)**

Anatomy of a Knowledge Graph



- ✓ Intuitive (e.g., no “foreign keys”)
- ✓ **Data** + **schema (ontology)**
- ✓ URIs not strings

Anatomy of a Knowledge Graph



Architectural Structure

Name	Height	Location
Eiffel Tower	324	Paris
Shard	310	London

Tower

Name
Eiffel Tower

Building

Name
Shard

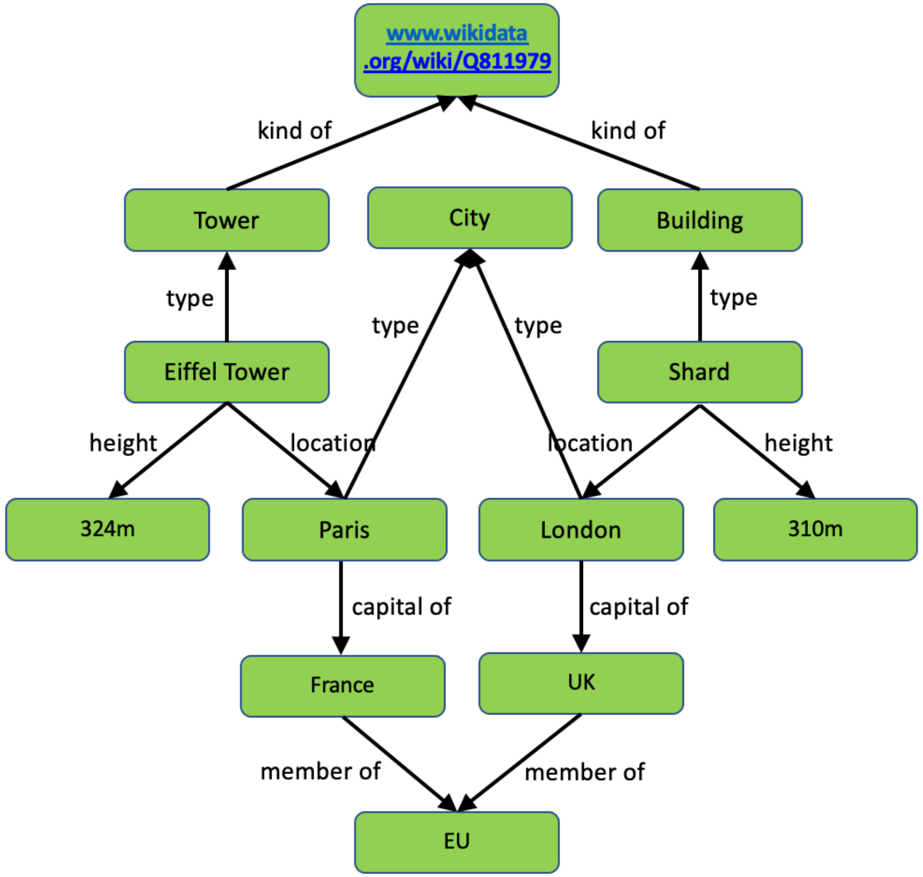
City

Name	Capital Of
Paris	France
London	UK

Member

Country	Union
France	EU
UK	EU

Anatomy of a Knowledge Graph



Architectural Structure		
Name	Height	Location
Eiffel Tower	324	Paris
Shard	310	London

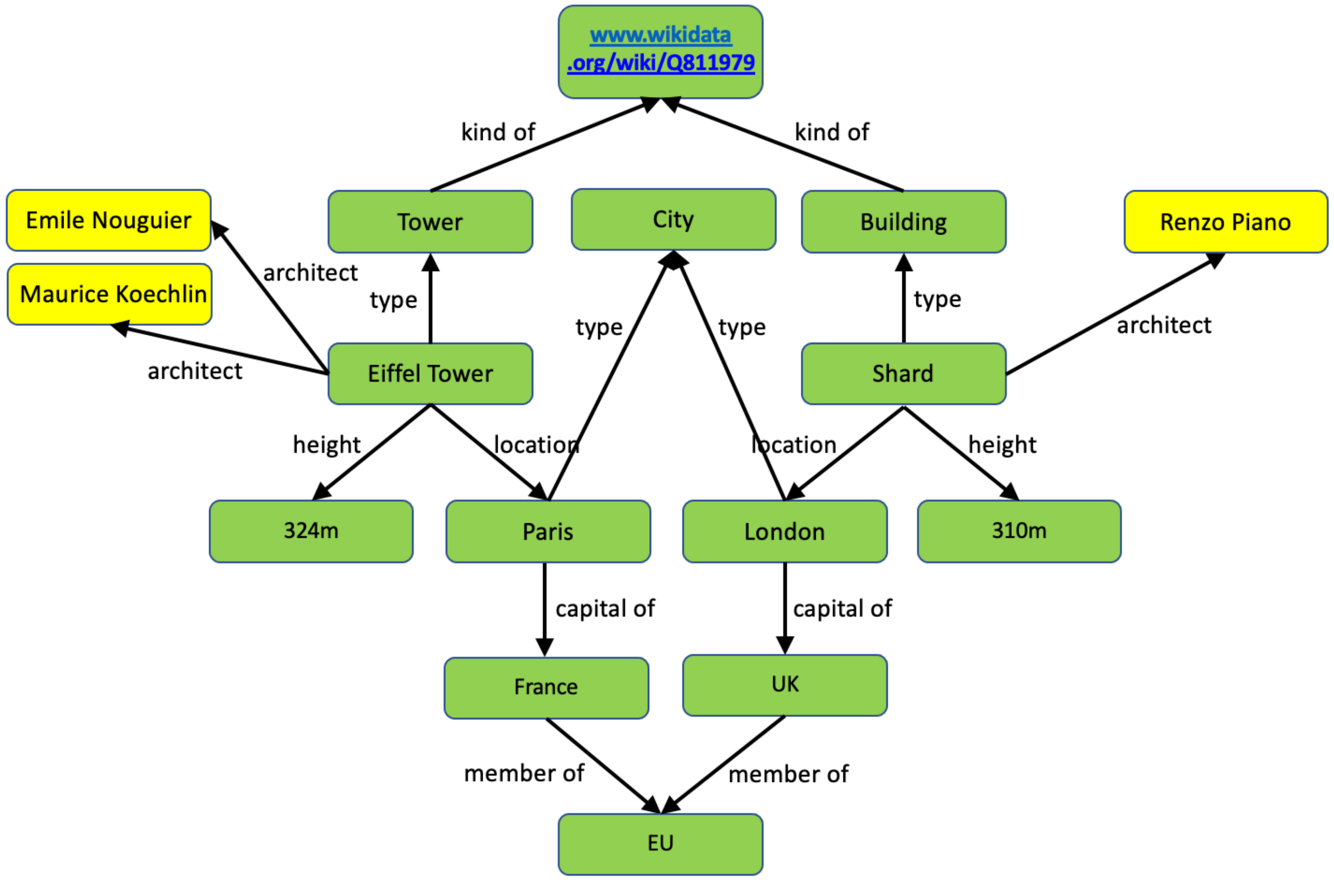
Tower
Name
Eiffel Tower

Building
Name
Shard

City	
Name	Capital Of
Paris	France
London	UK

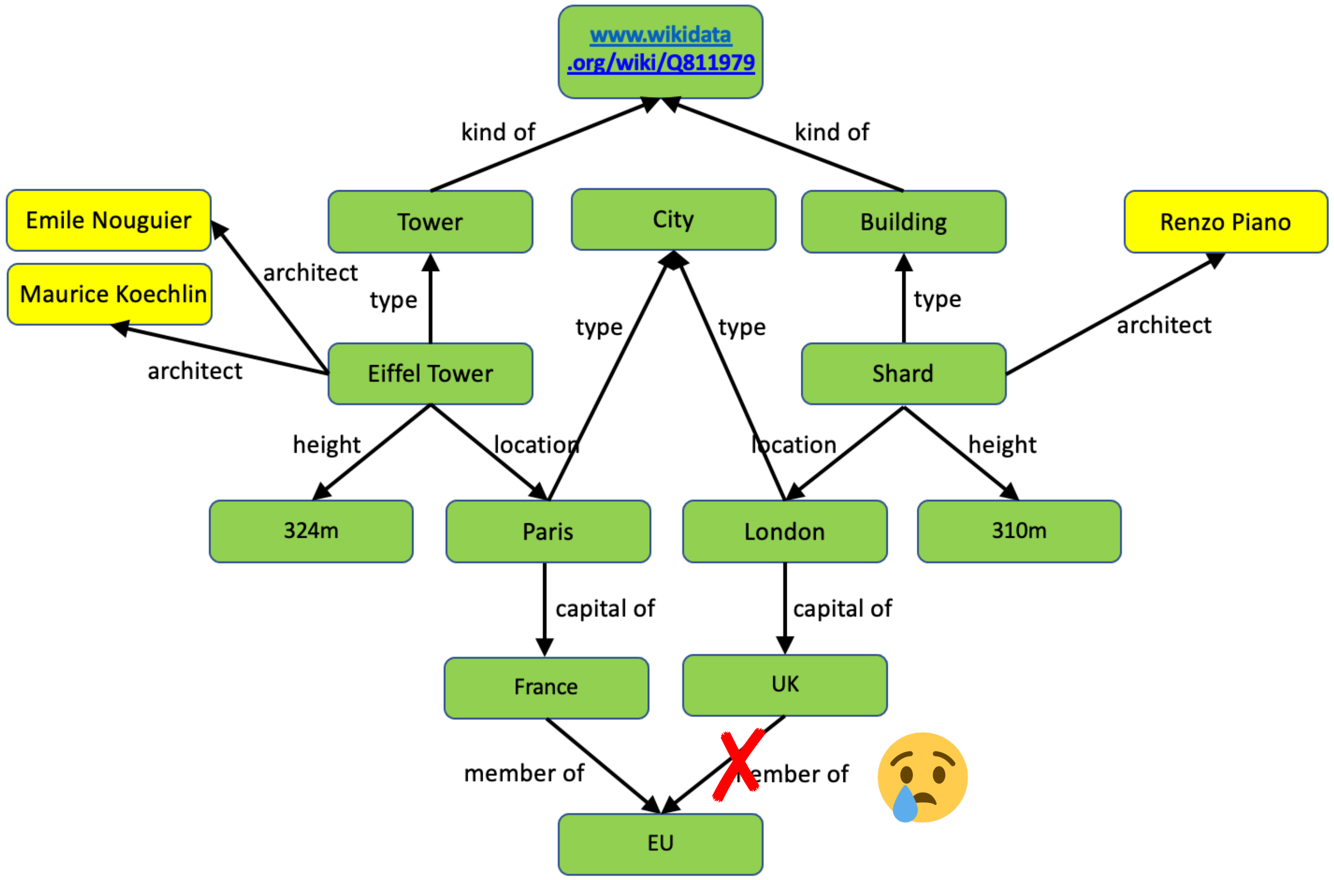
Member	
Country	Union
France	EU
UK	EU

Anatomy of a Knowledge Graph



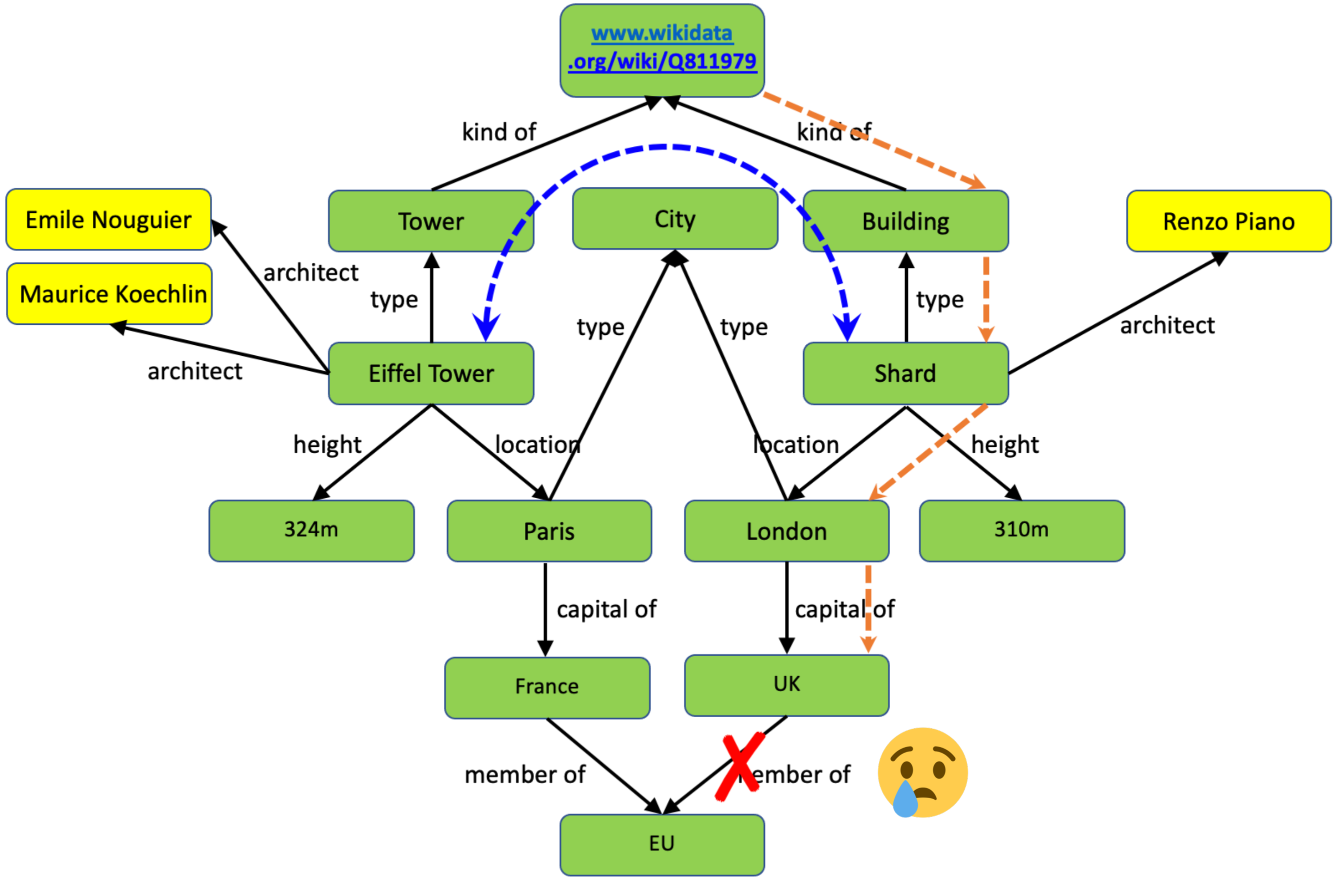
- ✓ Intuitive (e.g., no "foreign keys")
- ✓ **Data** + **schema (ontology)**
- ✓ URIs not strings
- ✓ Flexible & extensible

Anatomy of a Knowledge Graph



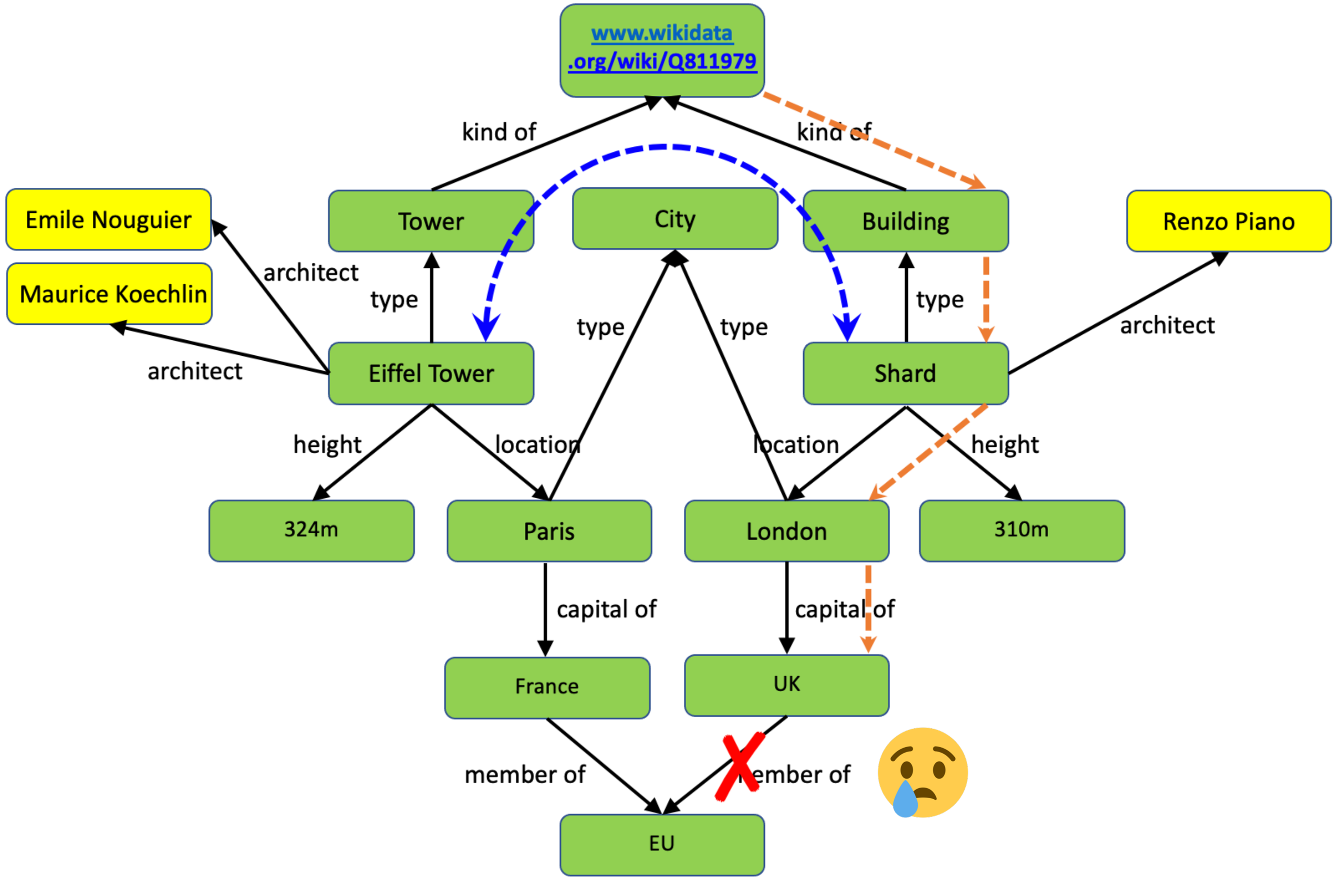
- ✓ Intuitive (e.g., no “foreign keys”)
- ✓ **Data** + **schema (ontology)**
- ✓ URIs not strings
- ✓ Flexible & extensible

Anatomy of a Knowledge Graph



- ✓ Intuitive (e.g., no "foreign keys")
- ✓ **Data** + **schema (ontology)**
- ✓ URIs not strings
- ✓ Flexible & extensible
- ✓ Other kinds of query
  - navigation
  - similarity & locality

Anatomy of a Knowledge Graph

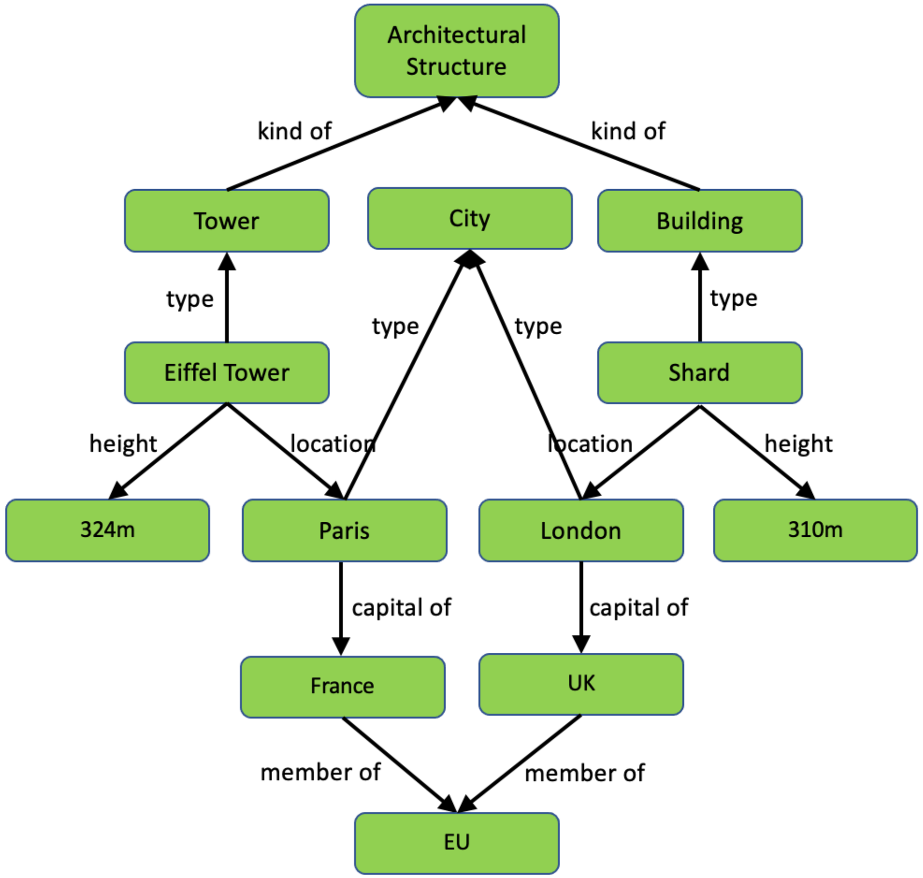


- ✓ Intuitive (e.g., no "foreign keys")
- ✓ **Data** + **schema (ontology)**
- ✓ URIs not strings
- ✓ Flexible & extensible
- ✓ Other kinds of query
  - navigation
  - similarity & locality

✗ Views
- Data integration & restructuring
- Security
- Query simplification & optimization
- …

Knowledge Graph Semantics

Knowledge Graph Semantics



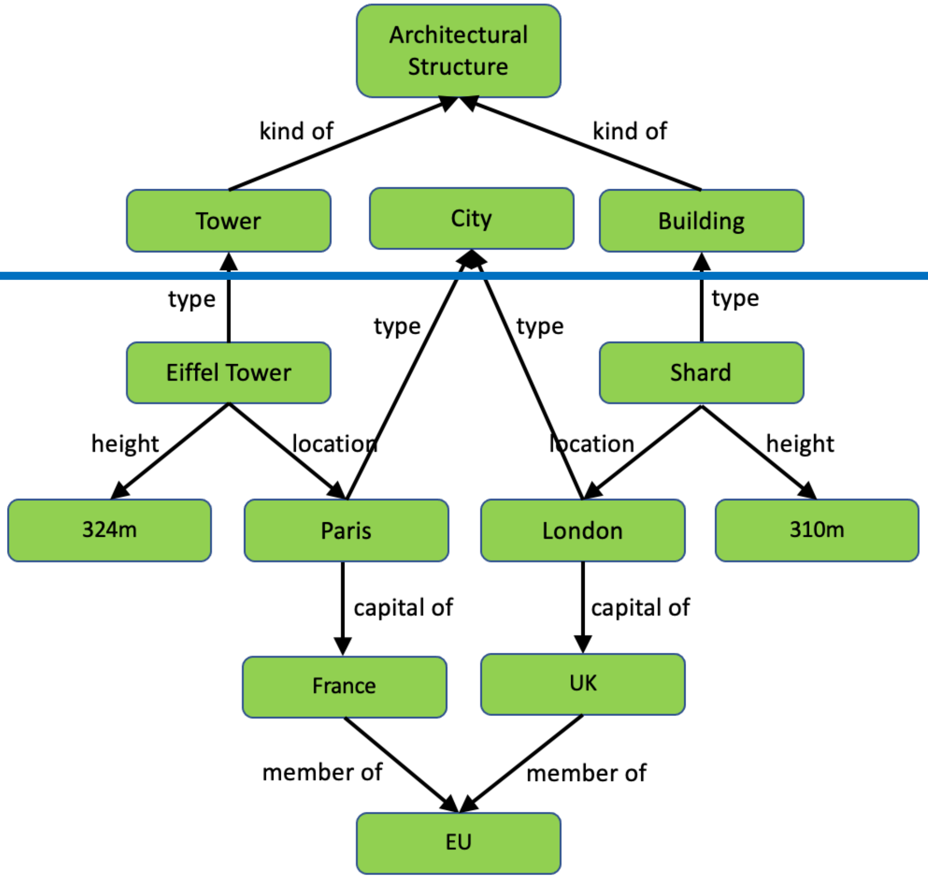
Why do we need semantics?

- To tell us how to use KG
- E.g., how to answer queries:
  - Architectural Structures with location in the EU?

Knowledge Graph Semantics

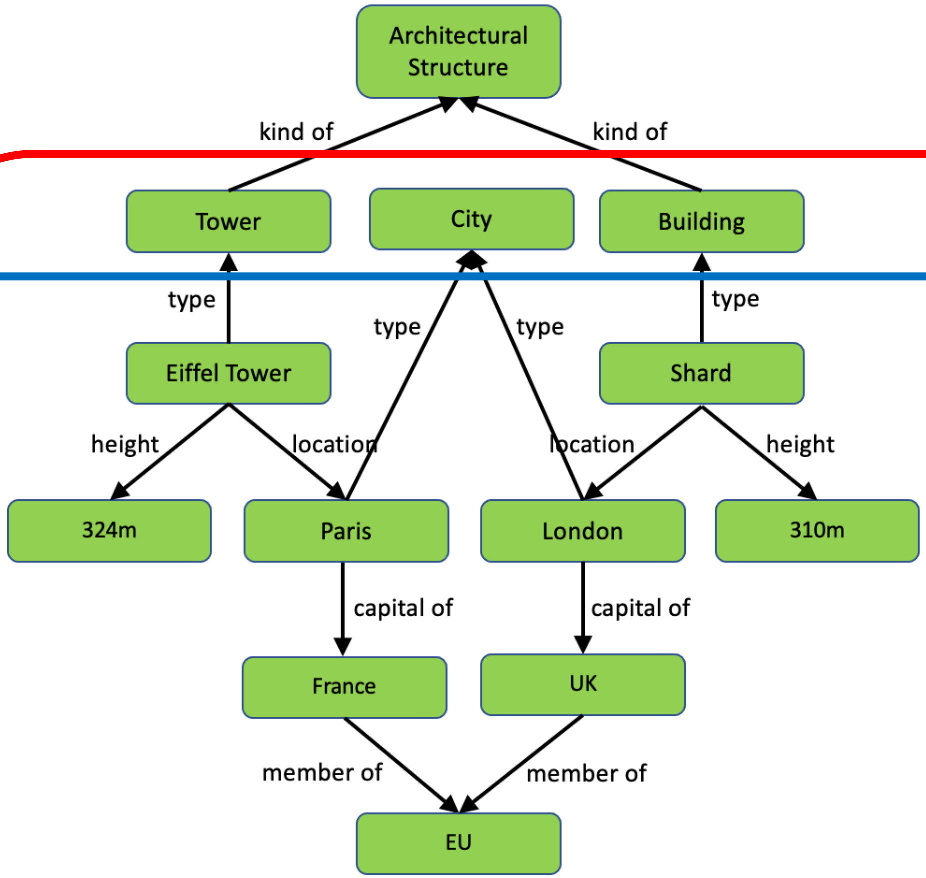
(OWL) ontology / conceptual schema

$$\forall x\ \mathsf{Tower}(x) \rightarrow \mathsf{ArchitecturalStructure}(x)$$
$$\forall x\ \mathsf{Building}(x) \rightarrow \mathsf{ArchitecturalStructure}(x)$$



Knowledge Graph Semantics

(OWL) ontology / conceptual schema



$\forall x \ \text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\forall x \ \text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$

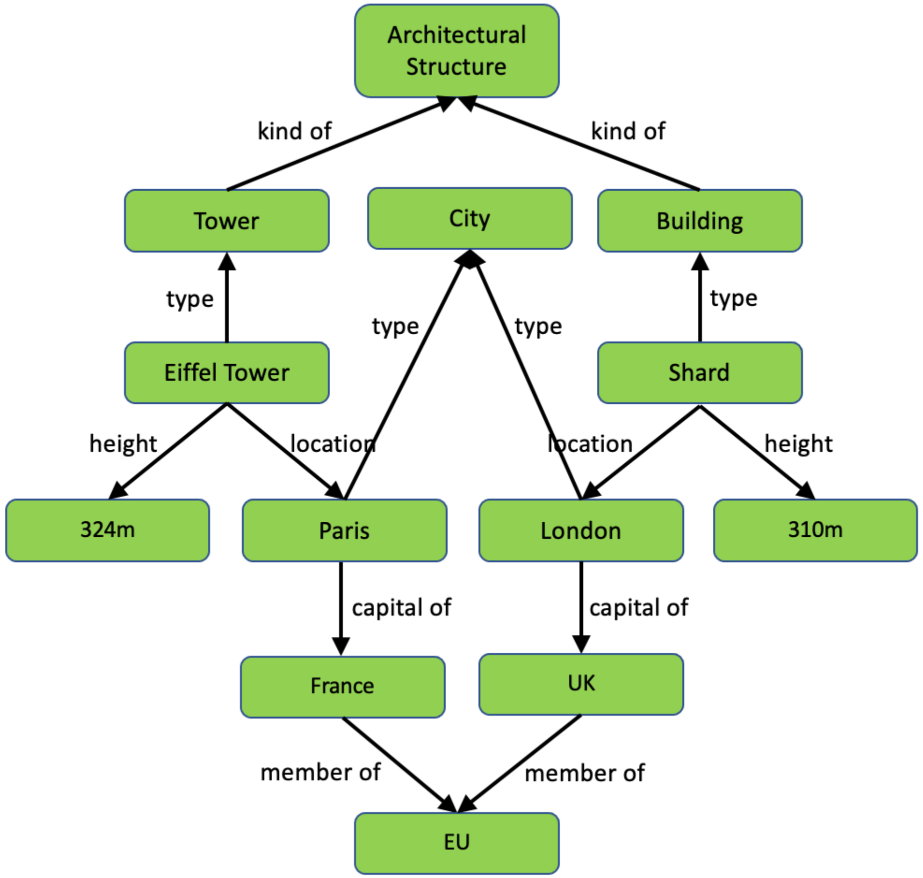
Tower(EiffelTower)
City(Paris)
location(EiffelTower, Paris)
location(Shard, London)
capital_of(Paris, France)
member_of(France, EU)

Building(Shard)
City(London)
height(EiffelTower, 324m)
height(Shard, 310m)
capital_of(London, UK)
member_of(UK, EU)

(RDF) graph / facts / data

Knowledge Graph Semantics

Knowledge base/graph

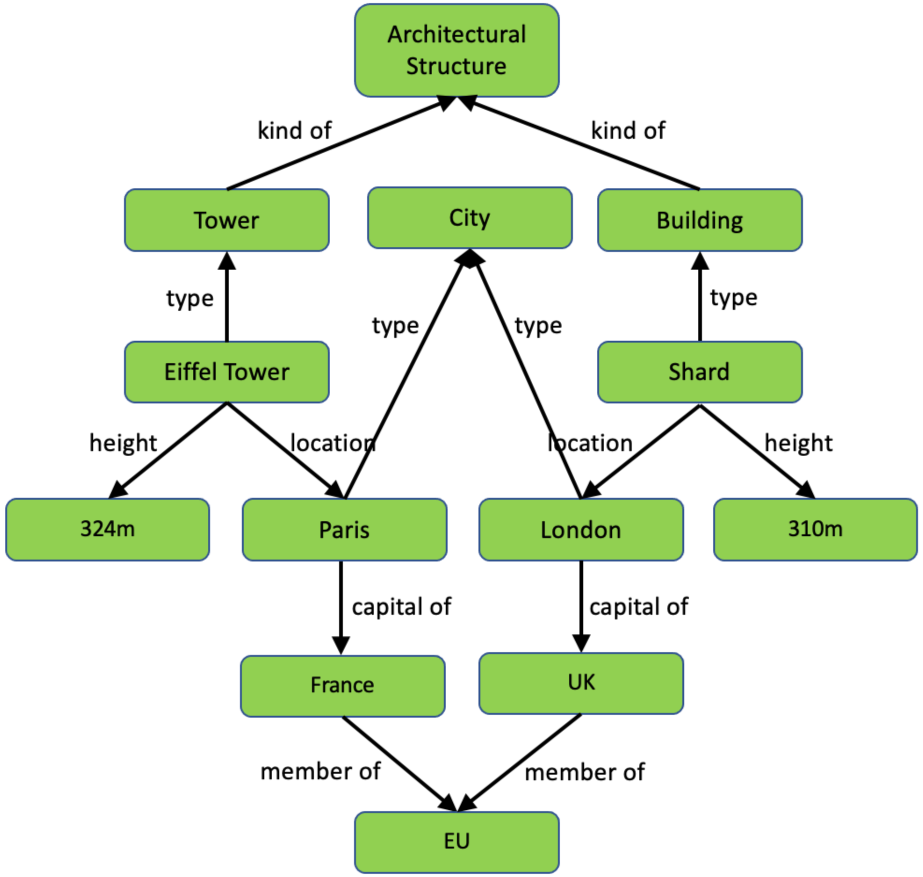


$\forall x$ Tower$(x) \rightarrow$ ArchitecturalStructure(x)
$\forall x$ Building$(x) \rightarrow$ ArchitecturalStructure(x)

Tower(EiffelTower)	Building(Shard)
City(Paris)	City(London)
location(EiffelTower, Paris)	height(EiffelTower, 324m)
location(Shard, London)	height(Shard, 310m)
capital_of(Paris, France)	capital_of(London, UK)
member_of(France, EU)	member_of(UK, EU)

Knowledge Graph Rules

Knowledge base/graph



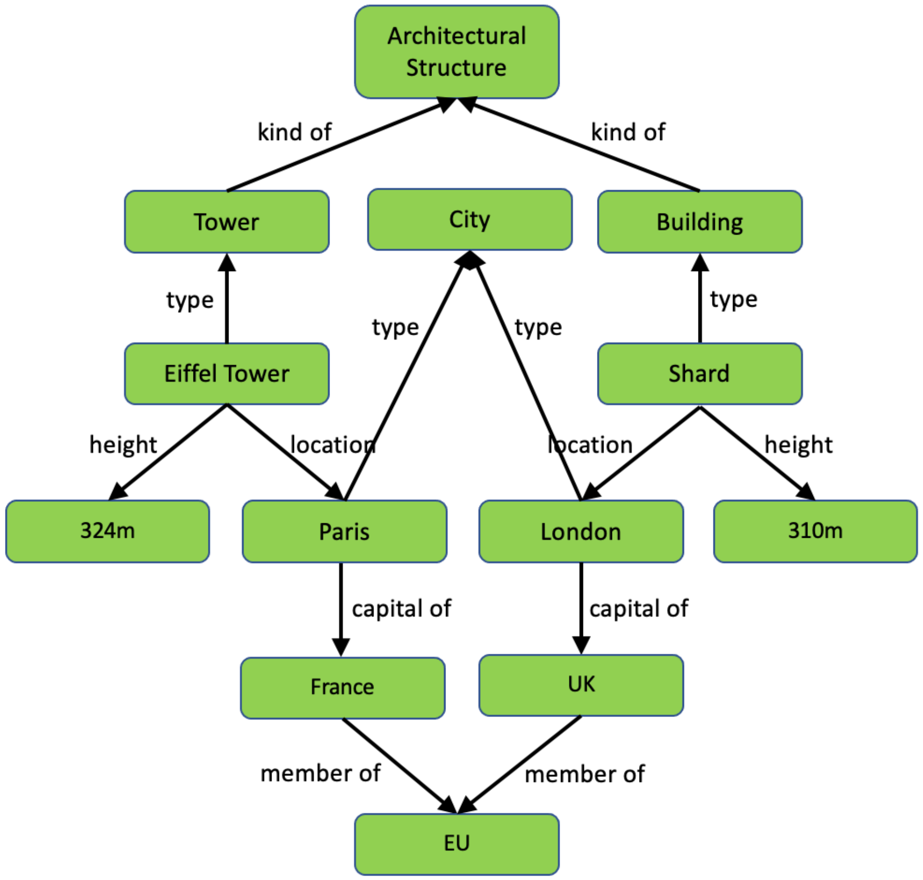
$\forall x \;\text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\forall x \;\text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\forall x, y, z \;\text{location}(x, y) \wedge \text{capital\_of}(y, z) \rightarrow \text{location}(x, z)$
$\forall x, y, z \;\text{location}(x, y) \wedge \text{member\_of}(y, z) \rightarrow \text{location}(x, z)$

$\text{Tower}(\text{EiffelTower})$
$\text{City}(\text{Paris})$
$\text{location}(\text{EiffelTower}, \text{Paris})$
$\text{location}(\text{Shard}, \text{London})$
$\text{capital\_of}(\text{Paris}, \text{France})$
$\text{member\_of}(\text{France}, \text{EU})$

$\text{Building}(\text{Shard})$
$\text{City}(\text{London})$
$\text{height}(\text{EiffelTower}, 324\text{m})$
$\text{height}(\text{Shard}, 310\text{m})$
$\text{capital\_of}(\text{London}, \text{UK})$
$\text{member\_of}(\text{UK}, \text{EU})$

Knowledge Graph Rules

Knowledge base/graph

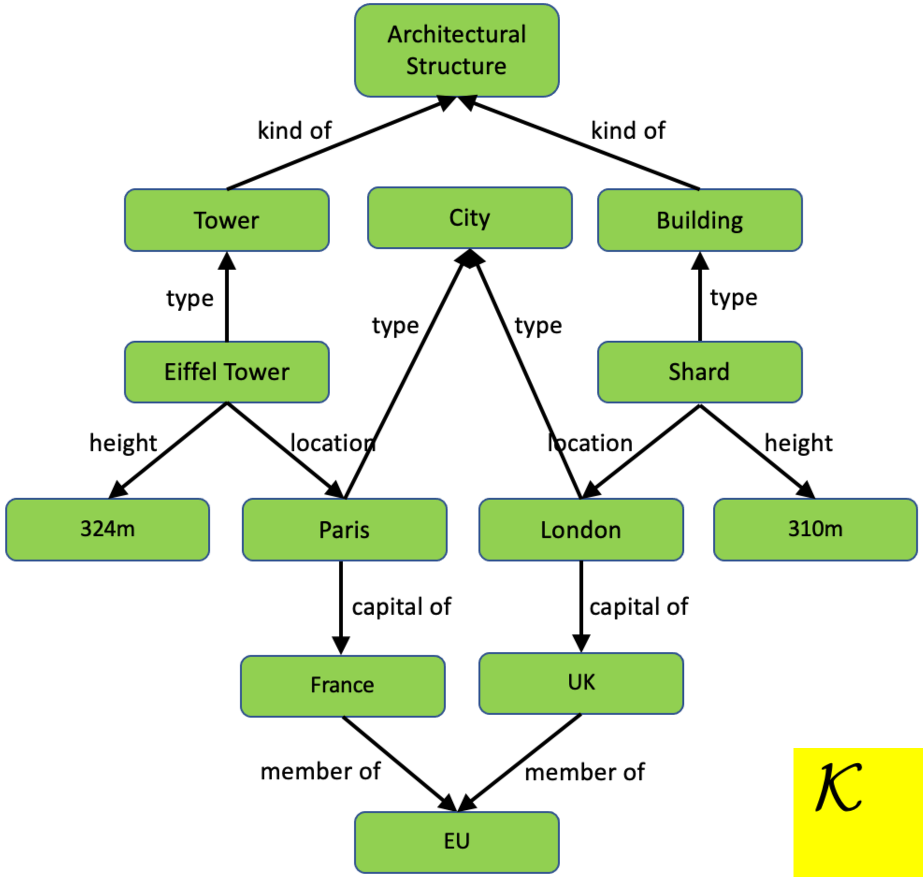


$\text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{location}(x, y) \wedge \text{capital\_of}(y, z) \rightarrow \text{location}(x, z)$
$\text{location}(x, y) \wedge \text{member\_of}(y, z) \rightarrow \text{location}(x, z)$

Tower(EiffelTower)	Building(Shard)
City(Paris)	City(London)
location(EiffelTower, Paris)	height(EiffelTower, 324m)
location(Shard, London)	height(Shard, 310m)
capital_of(Paris, France)	capital_of(London, UK)
member_of(France, EU)	member_of(UK, EU)

Knowledge Graph Query Answering

Knowledge base/graph



$\mathsf{Tower}(x) \rightarrow \mathsf{ArchitecturalStructure}(x)$
$\mathsf{Building}(x) \rightarrow \mathsf{ArchitecturalStructure}(x)$
$\mathsf{location}(x,y) \wedge \mathsf{capital\_of}(y,z) \rightarrow \mathsf{location}(x,z)$
$\mathsf{location}(x,y) \wedge \mathsf{member\_of}(y,z) \rightarrow \mathsf{location}(x,z)$

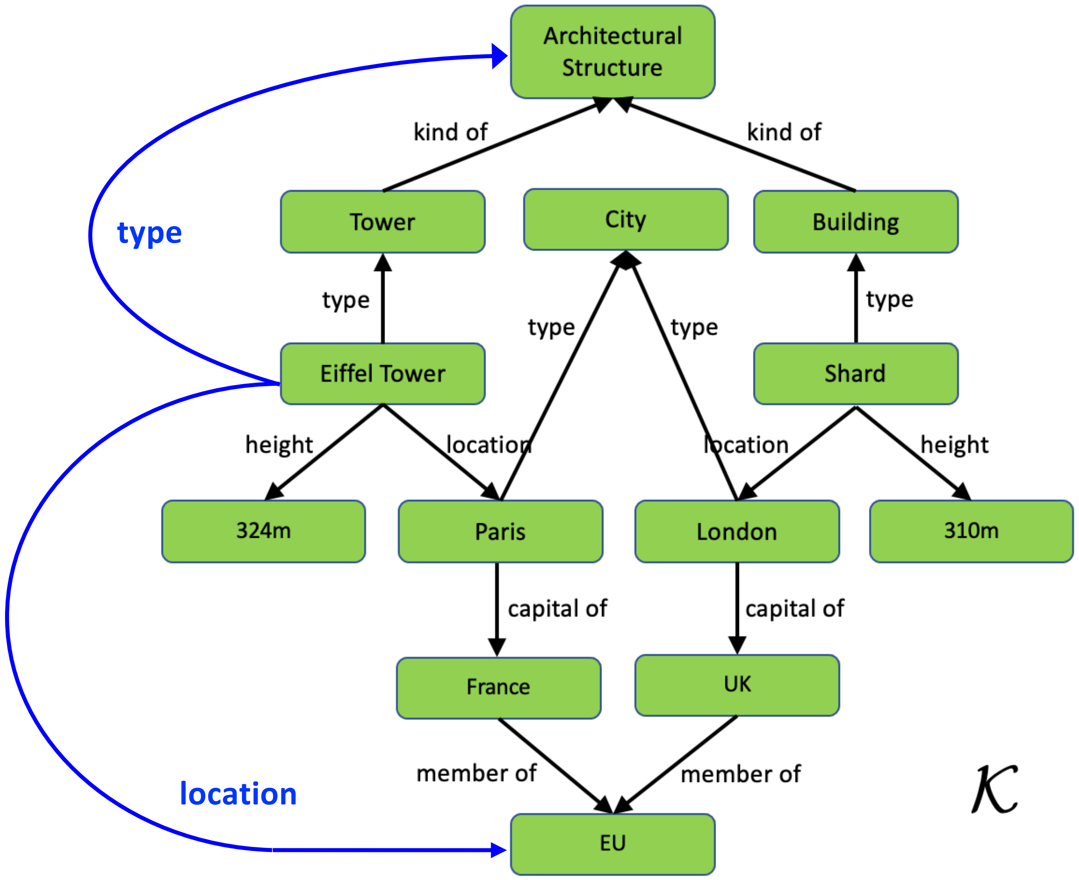
Tower(EiffelTower)
City(Paris)
location(EiffelTower, Paris)
location(Shard, London)
capital_of(Paris, France)
member_of(France, EU)

Building(Shard)
City(London)
height(EiffelTower, 324m)
height(Shard, 310m)
capital_of(London, UK)
member_of(UK, EU)

$\mathcal{K} \models \mathsf{ArchitecturalStructure}(\mathsf{EiffelTower}) \wedge \mathsf{location}(\mathsf{EiffelTower}, \mathsf{EU})$

Knowledge Graph Query Answering

Knowledge base/graph

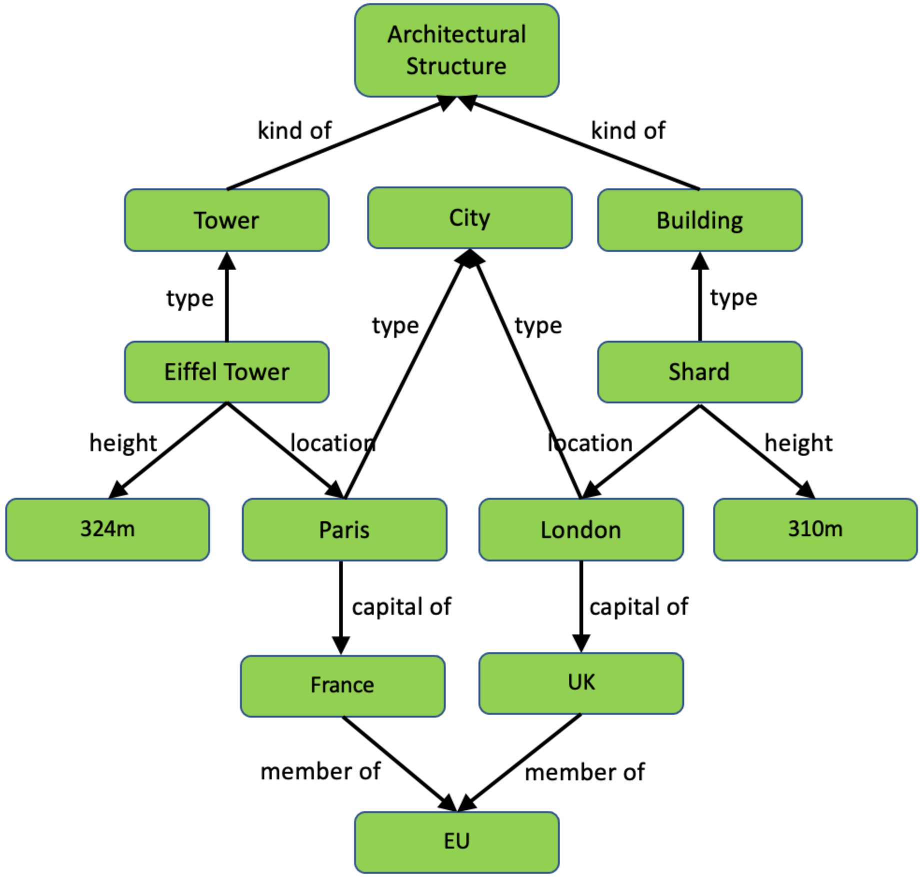


$\text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{location}(x, y) \wedge \text{capital\_of}(y, z) \rightarrow \text{location}(x, z)$
$\text{location}(x, y) \wedge \text{member\_of}(y, z) \rightarrow \text{location}(x, z)$

Tower(EiffelTower)	Building(Shard)
City(Paris)	City(London)
location(EiffelTower, Paris)	height(EiffelTower, 324m)
location(Shard, London)	height(Shard, 310m)
capital_of(Paris, France)	capital_of(London, UK)
member_of(France, EU)	member_of(UK, EU)

$$\mathcal{K} \models \text{ArchitecturalStructure}(\text{EiffelTower}) \wedge \text{location}(\text{EiffelTower}, \text{EU})$$

Rules and Views



$\text{Tower}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{Building}(x) \rightarrow \text{ArchitecturalStructure}(x)$
$\text{location}(x, y) \wedge \text{capital\_of}(y, z) \rightarrow \text{location}(x, z)$
$\text{location}(x, y) \wedge \text{member\_of}(y, z) \rightarrow \text{location}(x, z)$
$\text{ArchitecturalStructure}(x) \wedge \text{location}(x, EU) \rightarrow \text{EUStruc}(x)$

Views & Rules

- Integration & restructuring (e.g., introduce EUStruc)
- Security (e.g., only allow access to EUStruc)
- Simplification (e.g., use EUSruc in other queries/rules)
- Optimisation (e.g., materialize EUStruc)

Rules

- Recursive definitions (e.g., location)
- Critical for, e.g., part-whole, connectivity, causation, …

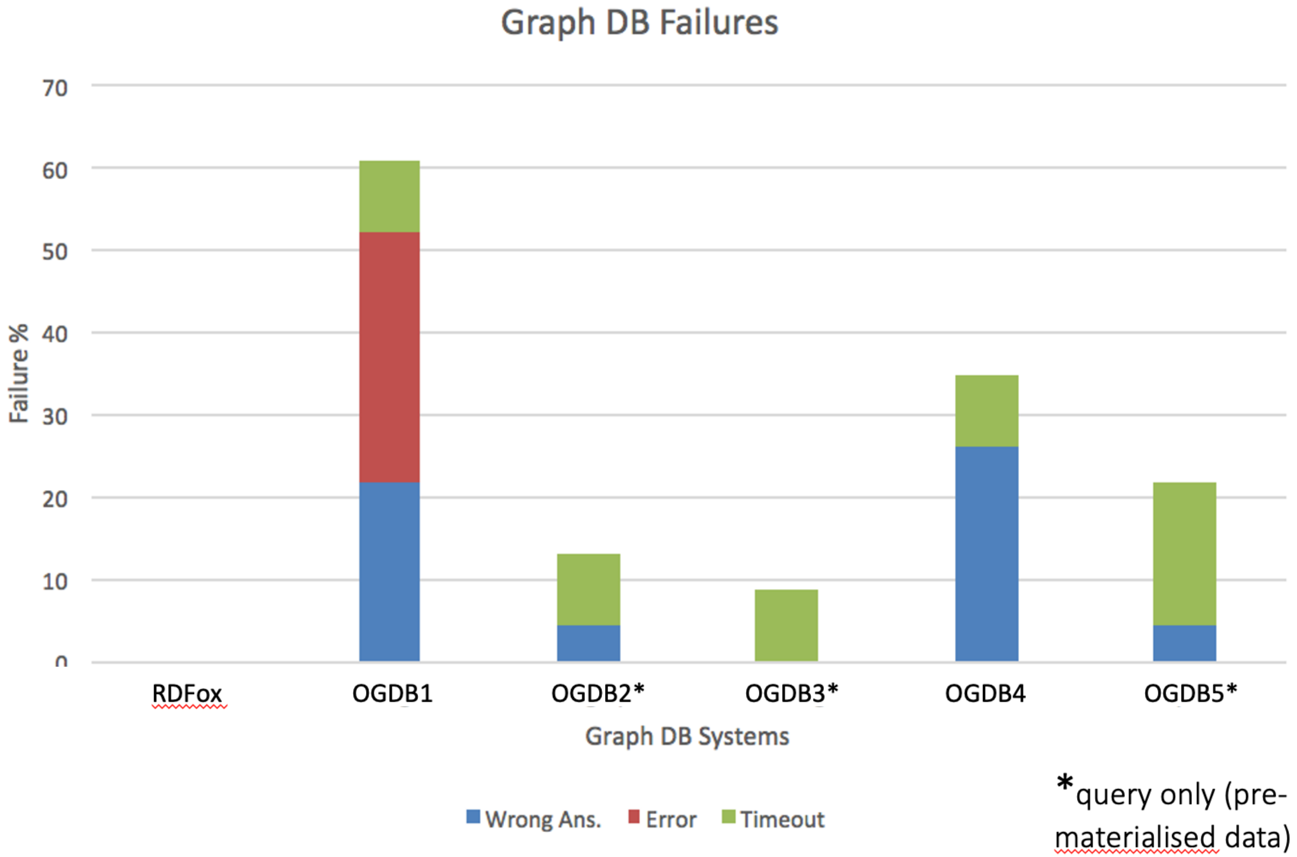
Knowledge Graph Systems

RDFOX

- Materialization reasoning seems ideal for data-centric applications
  - Can support expressive ontology/rule languages
  - Fast query answering over very large graphs
- Challenges
  - Materialisation can be costly in time and memory
  - How to deal with (rapidly) changing data
  - Reliability and correctness!
- Solution: RDFox
  - Optimised materialization exploiting modern multi-core architectures
  - Incremental maintenance as data changes
  - Formally specified and proven-correct algorithms

RDFOX

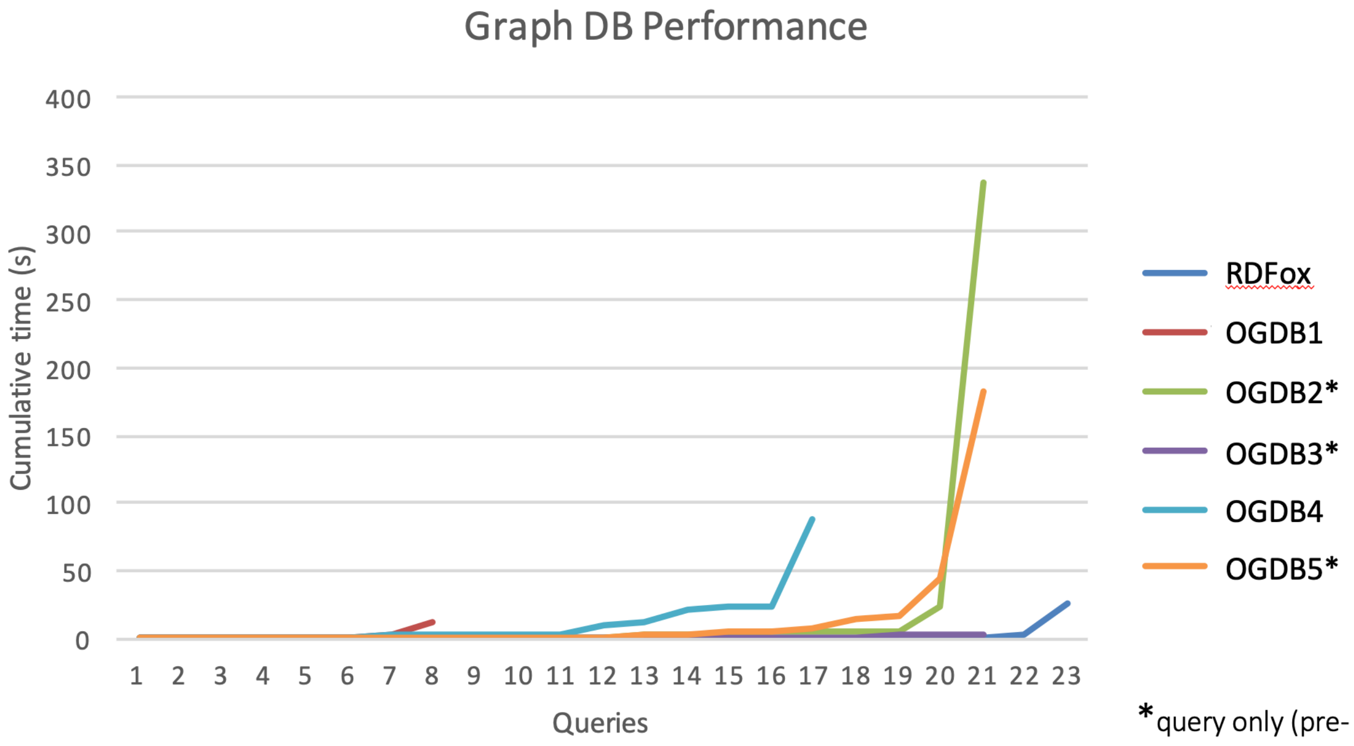
- **Novel algorithms developed at Oxford**
  - Proven correctness & performance



*query only (pre-materialised data)

RDFOX

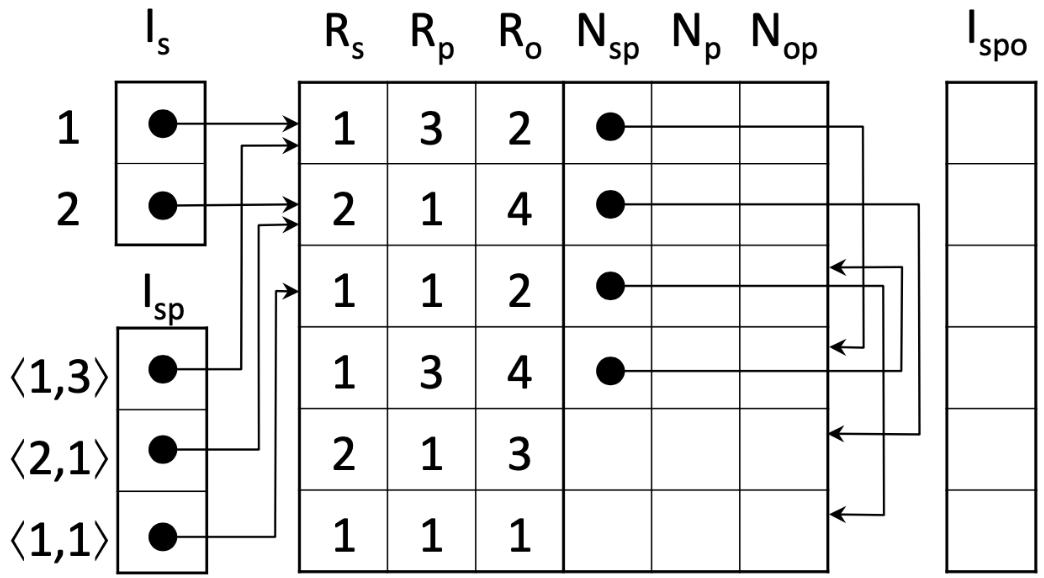
- **Novel algorithms developed at Oxford**
  - Proven correctness & performance



*query only (pre-materialised data)

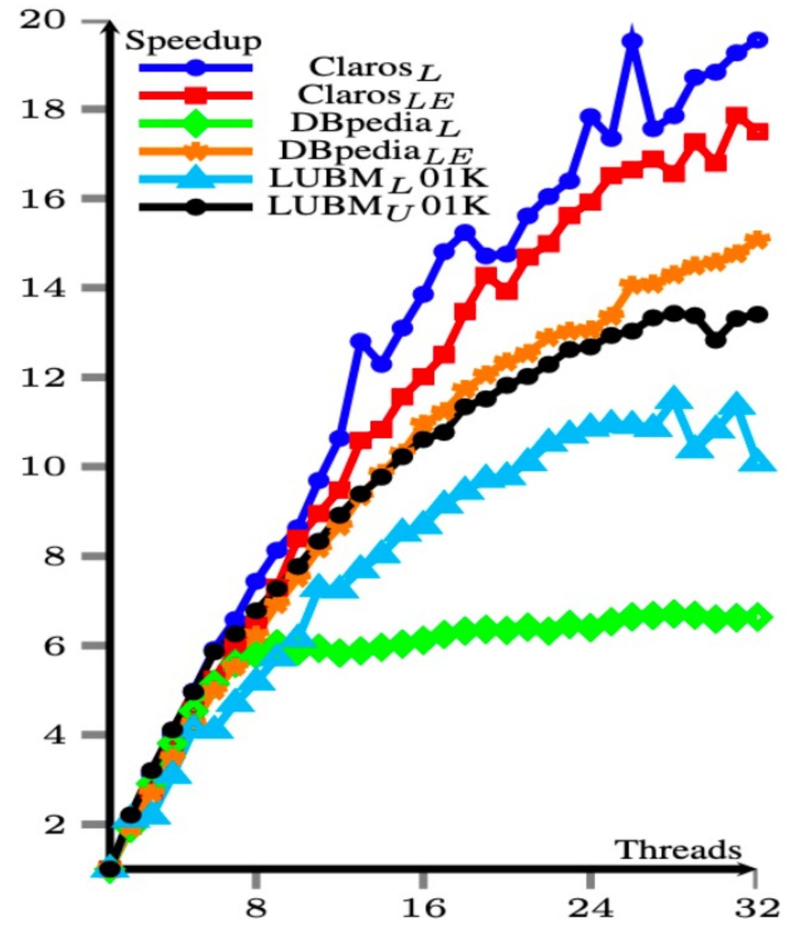
RDFOX

- **Novel algorithms developed at Oxford**
  - Proven correctness & performance
- **Optimized in-memory data structures**
  - $>10^{9}$ triples on 128 Gb entry level server
  - $>10^{10}$ triples on 1 Tb server



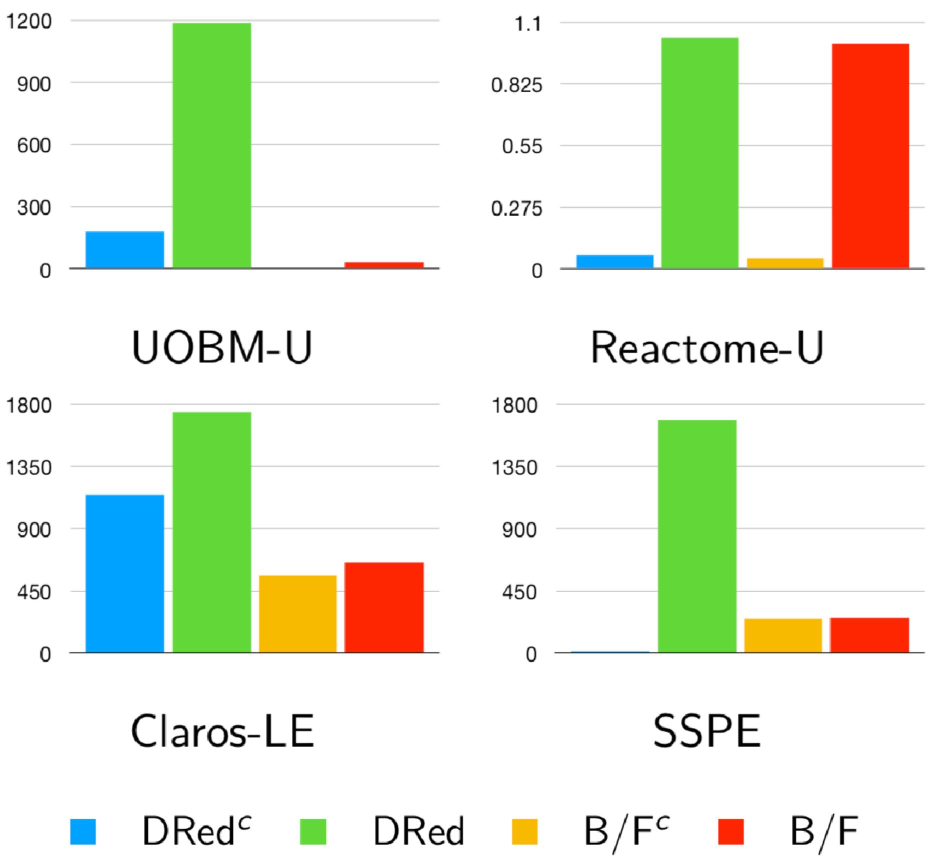
RDFOX

- **Novel algorithms developed at Oxford**
  - Proven correctness & performance
- **Optimized in-memory data structures**
  - >10^9 triples on 128 Gb entry level server
  - >10^{10} triples on 1 Tb server
- **Parallelised materialisation**
  - Dynamic distribution of workload
  - Mostly lock-free data structures

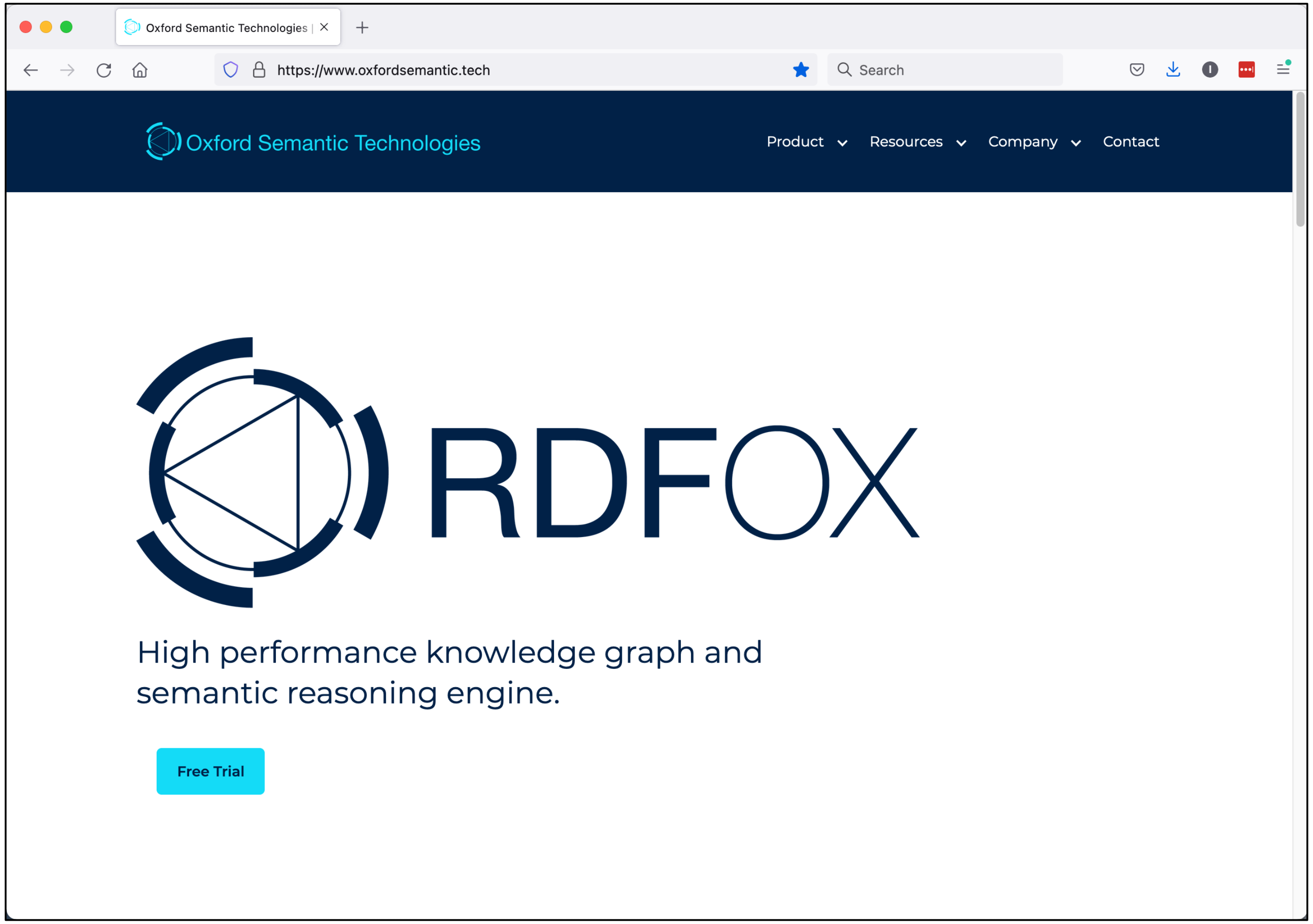


RDFOX

- **Novel algorithms developed at Oxford**
  - Proven correctness & performance
- **Optimized in-memory data structures**
  - $>10^9$ triples on 128 Gb entry level server
  - $>10^{10}$ triples on 1 Tb server
- **Parallelised materialisation**
  - Dynamic distribution of workload
  - Mostly lock-free data structures
- **Incremental addition and retraction**
  - Novel B/F materialisation maintenance algorithm

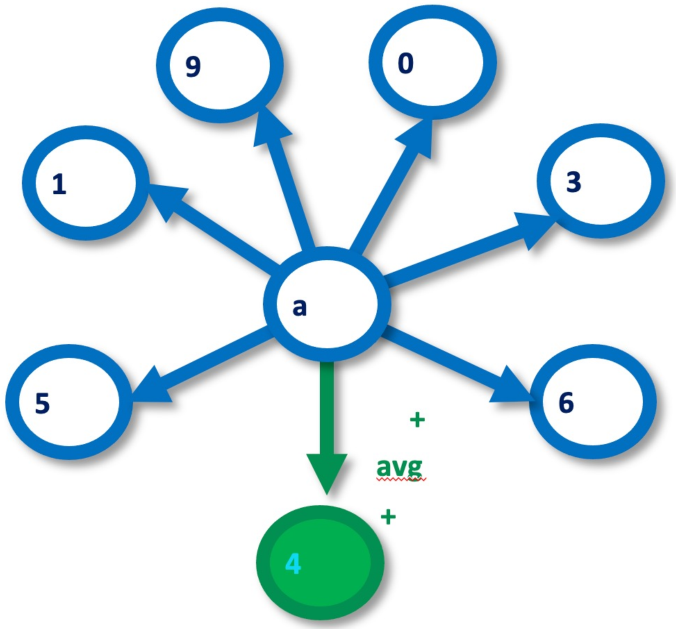


Oxford Semantic Technologies

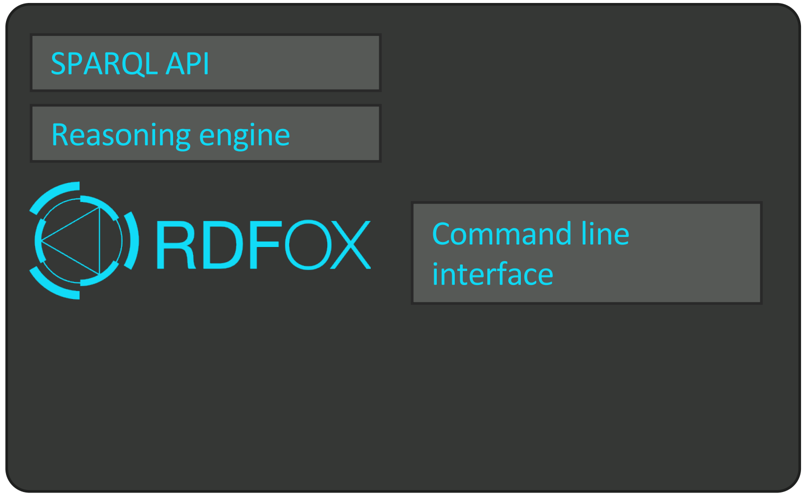


Extensions (beyond OWL RL)

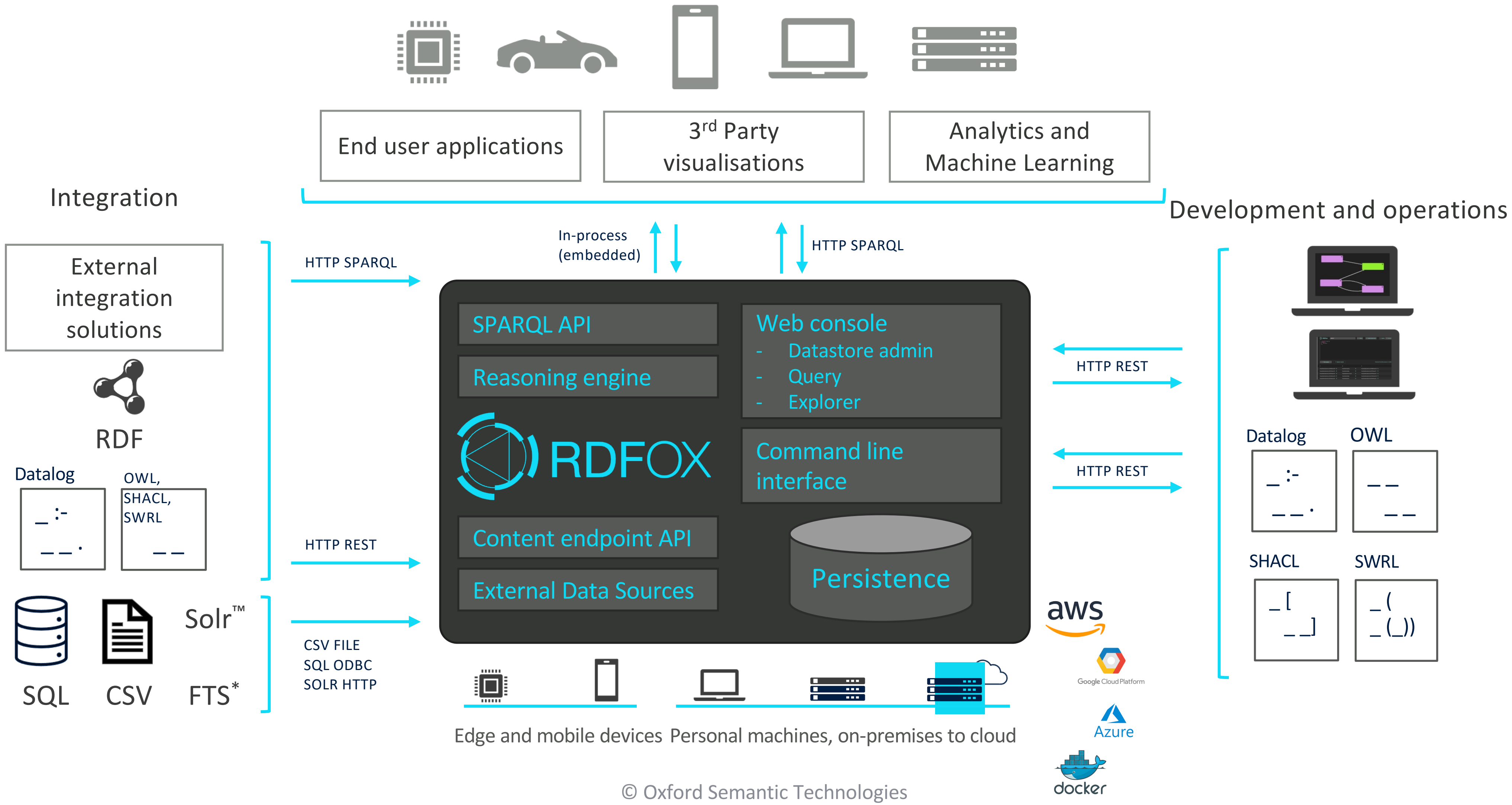
- Arbitrary rules
  - No restriction to OWL RL (tree-shaped) rules
- Data types and values
  - Numbers, strings, dates, …
  - Built in functions and aggregation
- Value invention
  - Add new (possibly computed) values to graph
  - Add new URI nodes to graph
- Constraints and negation as failure
  - SHACL+



System Architecture



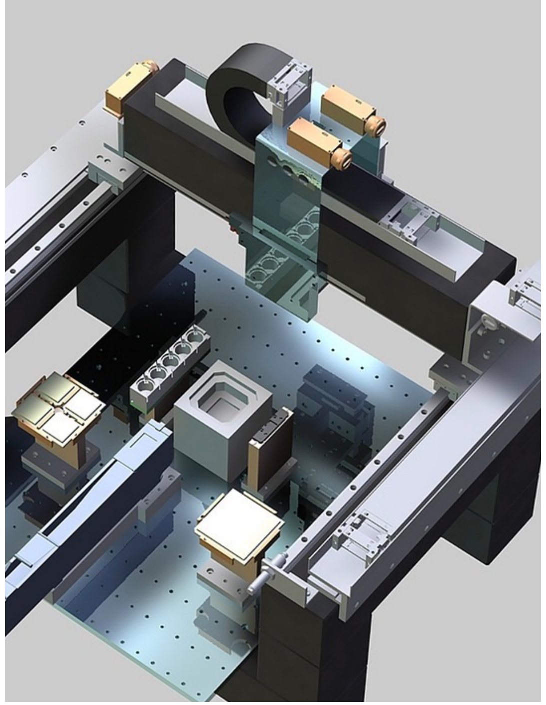
System Architecture

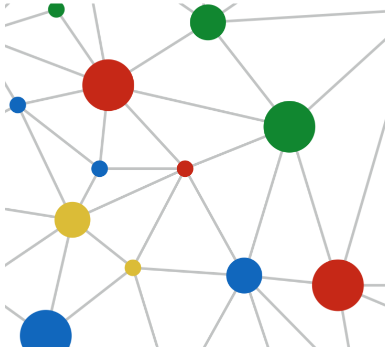


Knowledge Graph Applications

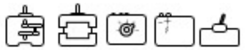
Configuration management

FESTO

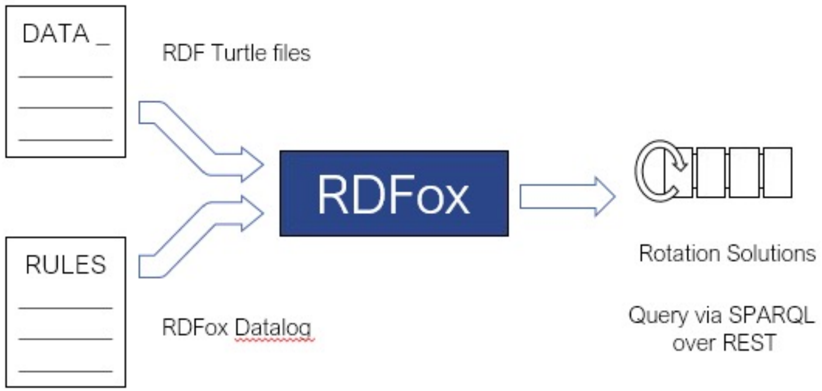


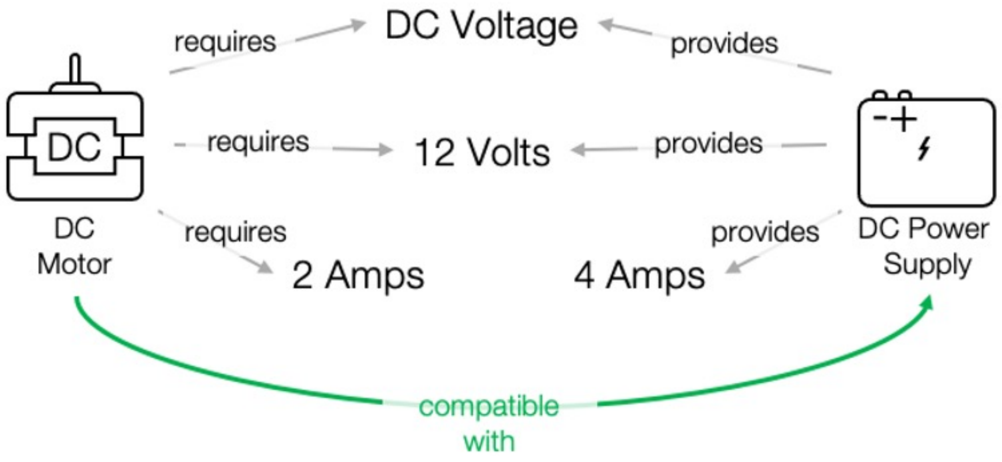


- Components
- Their attributes & constraints



- Definitions of compatibility &
- Valid configurations



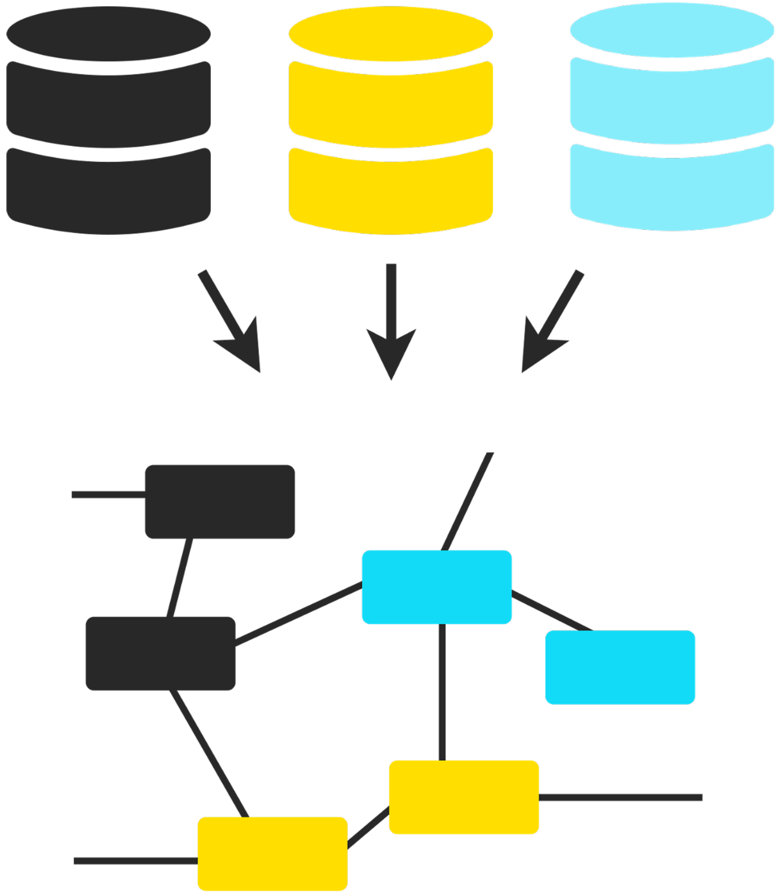



```
[?M, :compatibleWith, ?PS] :-
  :DCMotor[?M] ,
  :DCPowerSupply[?PS] ,
  [?PS, :provides, :DCVoltage] ,
  [?PS, :providedVoltage, ?pv] ,
  [?PS, :providedCurrent, ?pc] ,
  [?M,  :requires, :DCVoltage] ,
  [?M,  :requiredVoltage, ?rv] ,
  [?M,  :requiredCurrent, ?rc] ,
  FILTER (?pv = ?rv && ?pc >= ?rc) .
```

Data Integration

DOW JONES

- Integrate data from multiple sources
  - Companies
  - Executives
  - Stock markets
  - Geonames
  - Articles from WSJ, Factiva, …
- Query integrated data
  - Competitor companies that are NASDAQ listed and have subsidiaries in same or related sector
  - Article published between 2020-05-24 and 2020-05-26 that talk about company C and mention an African country



Wrap-up

Summary

- **KGs are powerful tool** for representing & reasoning about knowledge
- **Many applications**: configuration, data integration, fraud detection, …
- **Technical challenges**: scalability, correctness, **knowledge engineering** …
- **Solutions** based on **foundational research + systems engineering**

Thanks for Listening
Any Questions?

RDFOX : **www.oxfordsemantic.tech**

Background reading:

- **Description Logic:** Baader, Horrocks, Lutz, and Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017.
- **OWL:** Horrocks, Patel-Schneider, and van Harmelen. *From SHIQ and RDF to OWL: The Making of a Web Ontology Language*. J. of Web Semantics, 1(1):7-26, 2003.
- **RDFox algorithms & data structures:** Motik, Nenov, Piro, Horrocks, and Olteanu. *Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems*. AAAI 2014.
- **Incremental maintenance:** Motik, Nenov, Robert Piro, and Horrocks. *Maintenance of datalog materialisations revisited*. Artificial Intelligence, 269:76-136, 2019.