Using Semantic Technology to Tackle Industry's Data Variety Challenge

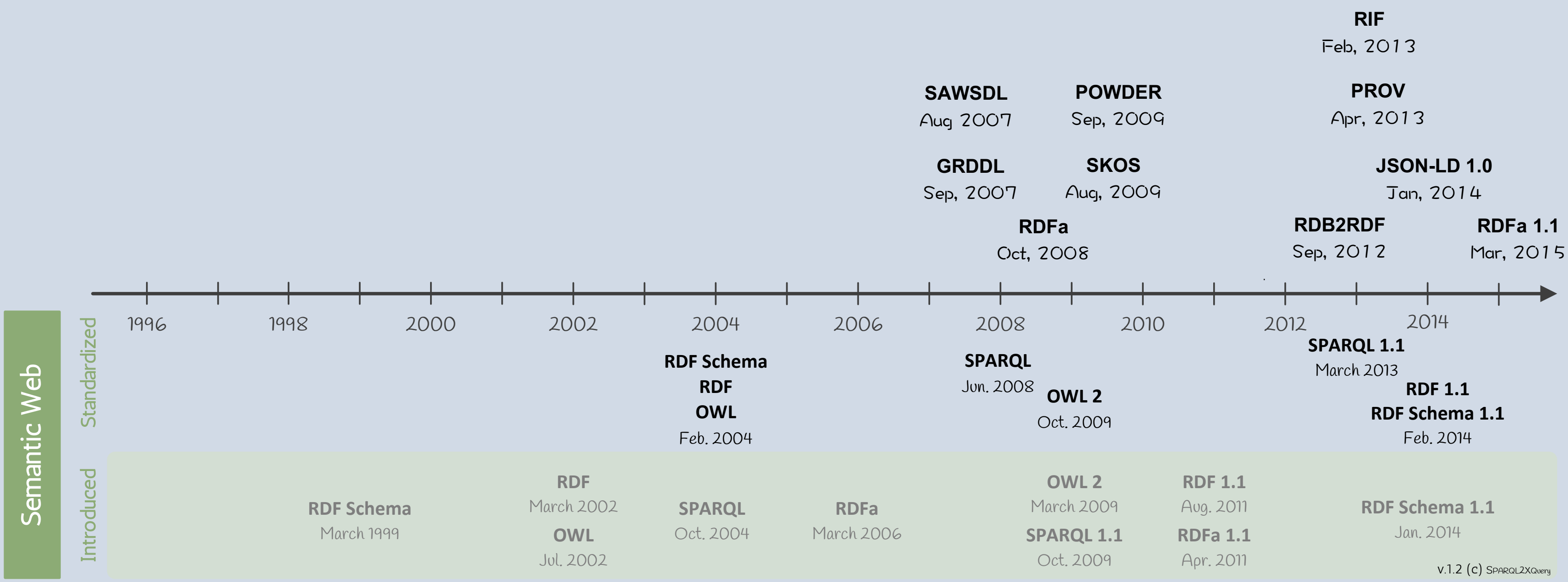
Ian Horrocks
Information Systems Group
Department of Computer Science
University of Oxford

大家好!

Genesis of Semantic Web

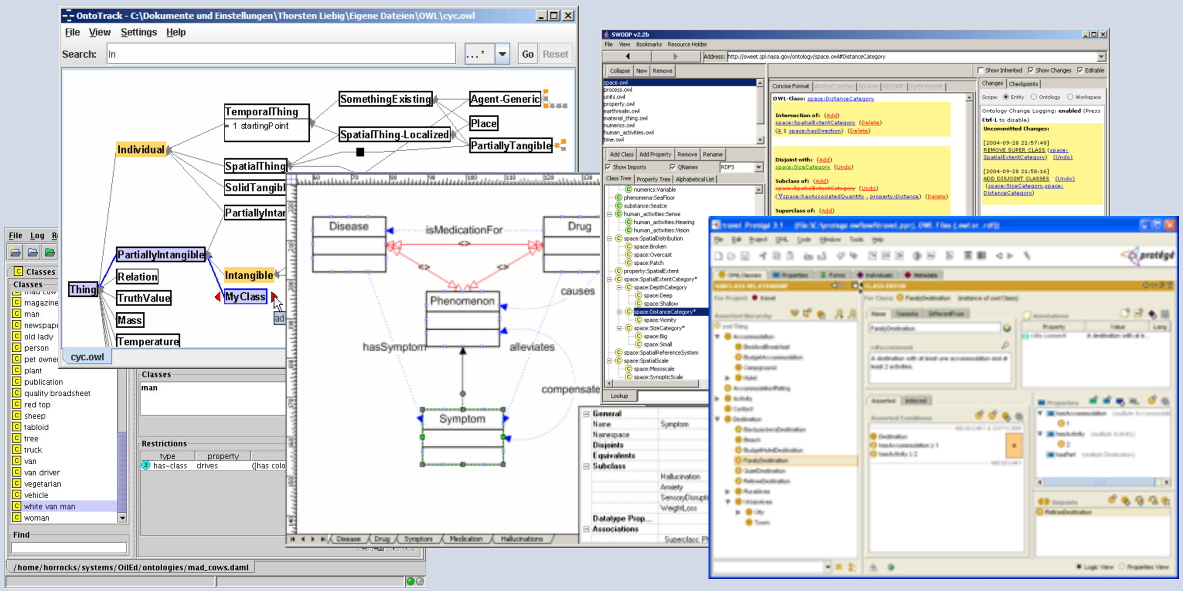
"A new form of Web content that is meaningful to computers will **unleash a revolution of new possibilities**"

Semantic Web Standards



University of Oxford Department of Computer Science
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique

Semantic Technology Infrastructure



FaCT++

pellet

Racer

Hermit

protégé

ORACLE

OWLIM

TrOWL
www.trowl.eu

semantic web framework
Jena

QuOnto
QUerying ONTOlogies

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

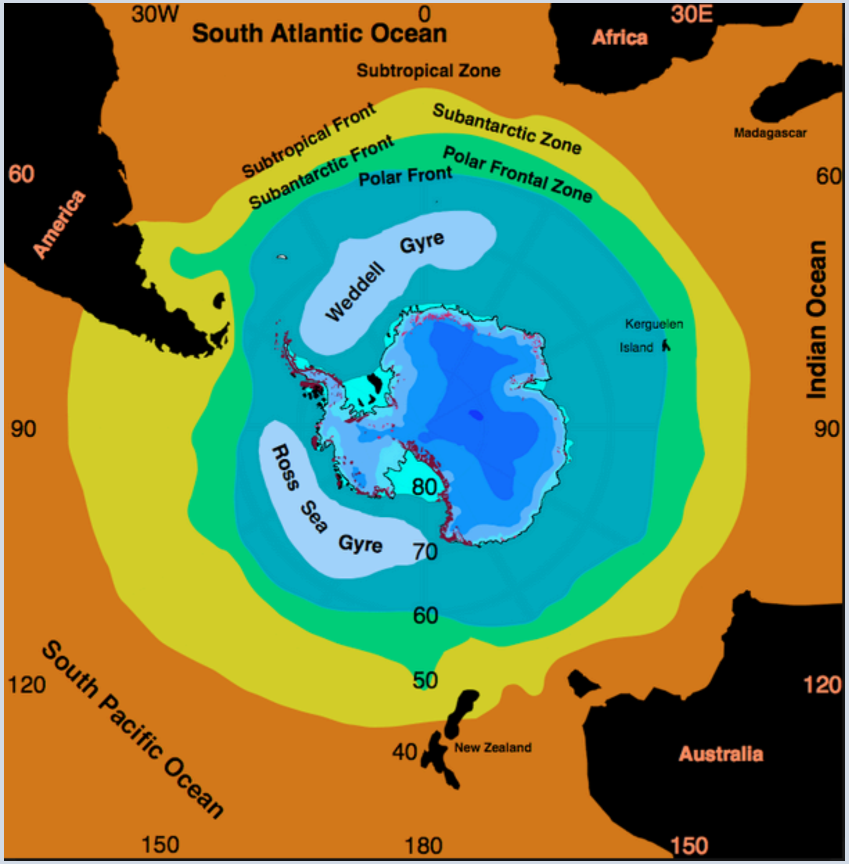
Information Systems Group

EPSRC
Engineering and Physical Sciences Research Council

Optique

Ontology-Centric Applications

- Agriculture
- Astronomy
- Oceanography
- Defence
- Education
- Energy management
- Geography
- Gioscience
- Life sciences
- …



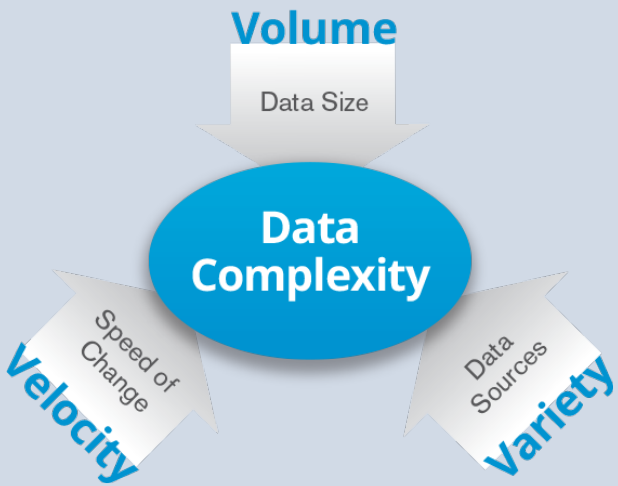
DEPARTMENT OF COMPUTER SCIENCE UNIVERSITY OF OXFORD
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique

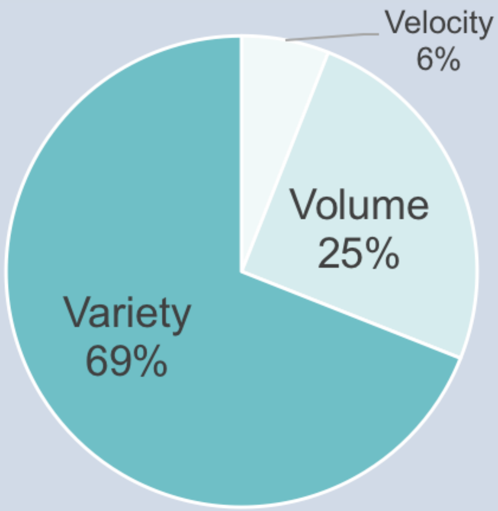
Ontology-Centric Applications

- **OBO foundry** includes more than 100 biological and biomedical ontologies
- **Siemens** "actively building OWL based clinical solutions"
- **SNOMED-CT** (Clinical Terms) ontology
  - used in healthcare systems of more than 15 countries, including Australia, Canada, Denmark, Spain, Sweden and the UK
  - also used by major US providers, e.g., Kaiser Permanente
  - ontology provides common vocabulary for recording clinical data

(Big) Data-Centric Applications

- *a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications* (WIKIPEDIA)
- Complexity due to range of factors including Volume, Velocity & Variety
- ***Variety***, *Not Volume, Is Driving Big Data Initiatives* (MITSloan Management Review)





DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique

Data Access: Statoil Exploration

- Geologists & geophysicists use data from previous operations in nearby locations to develop stratigraphic models of unexplored areas
  - TBs of relational data
  - using diverse schemata
  - spread over 1,000s of tables
  - and multiple data bases

Data Access

- 900 geologists & geophysicists
- 30-70% of time on data gathering
- 4 day turnaround for new queries

Data Exploitation

- Better use of experts time
- Data analysis “most important factor” for drilling success

Data Access: SIEMENS Energy Services

- Service centres responsible for remote monitoring and diagnostics of 1,000s of gas/steam turbines
- Engineers use a variety of data for visualization, diagnostics and trend detection:
  - several TB of time-stamped sensor data
  - several GB of event data
  - data grows at 30GB per day

Service Requests

- 1,000 requests per center per year
- 80% of time used on data gathering

Diagnostic Functionality

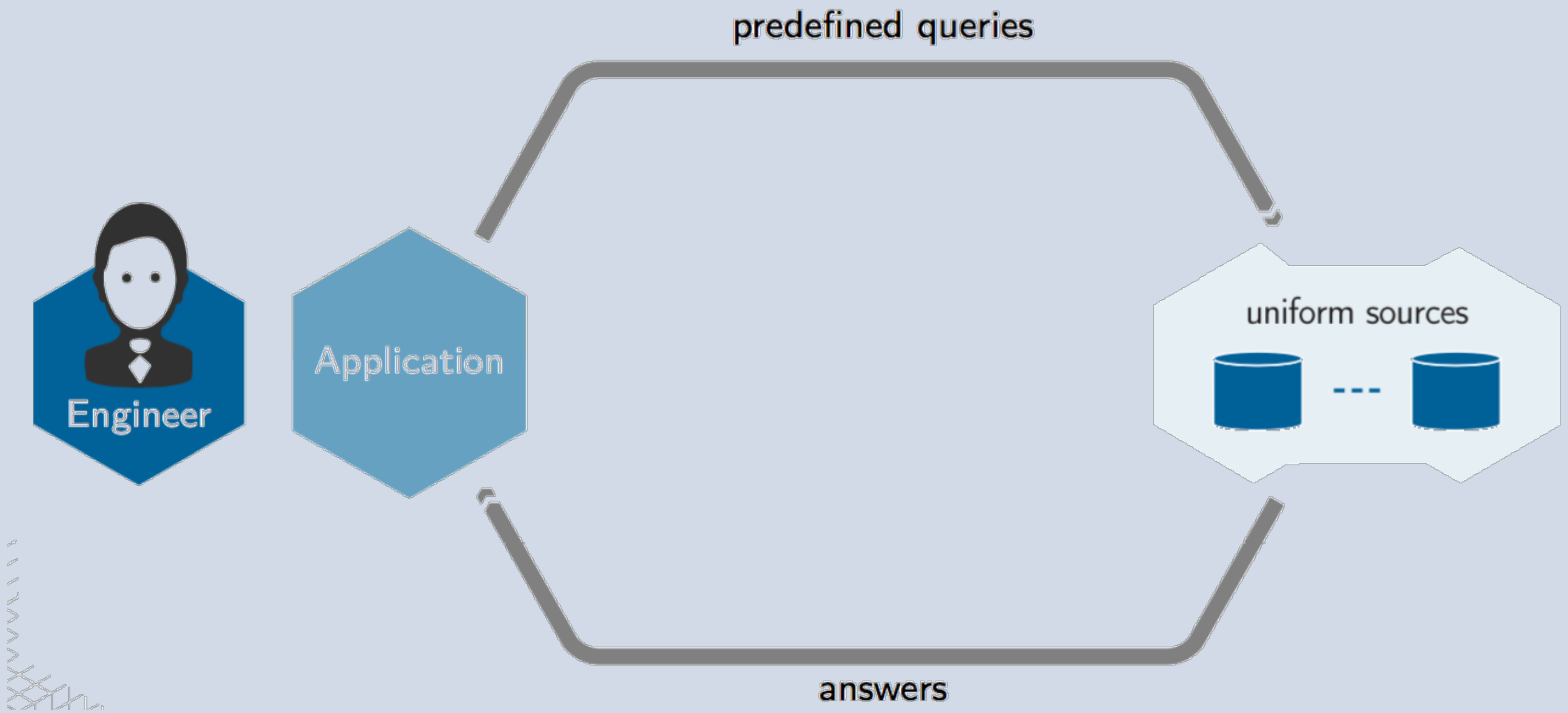
- 2–6 p/m to add new function
- New diagnostics → better exploitation of data

Data Access: Pervasive Problem

- **Oil and Gas** (e.g., Statoil, Schlumberger, …)
- **Power generation** (e.g., Siemens, …)
- **Power distribution** (e.g., EDF, …)
- **Engineering** (e.g., DNV, Aibel, …)
- **Finance** (e.g., Goldman Sachs, …)
- **Healthcare** (e.g., Kaiser Permanente, …)
- **Security** (e.g., “government agencies”)

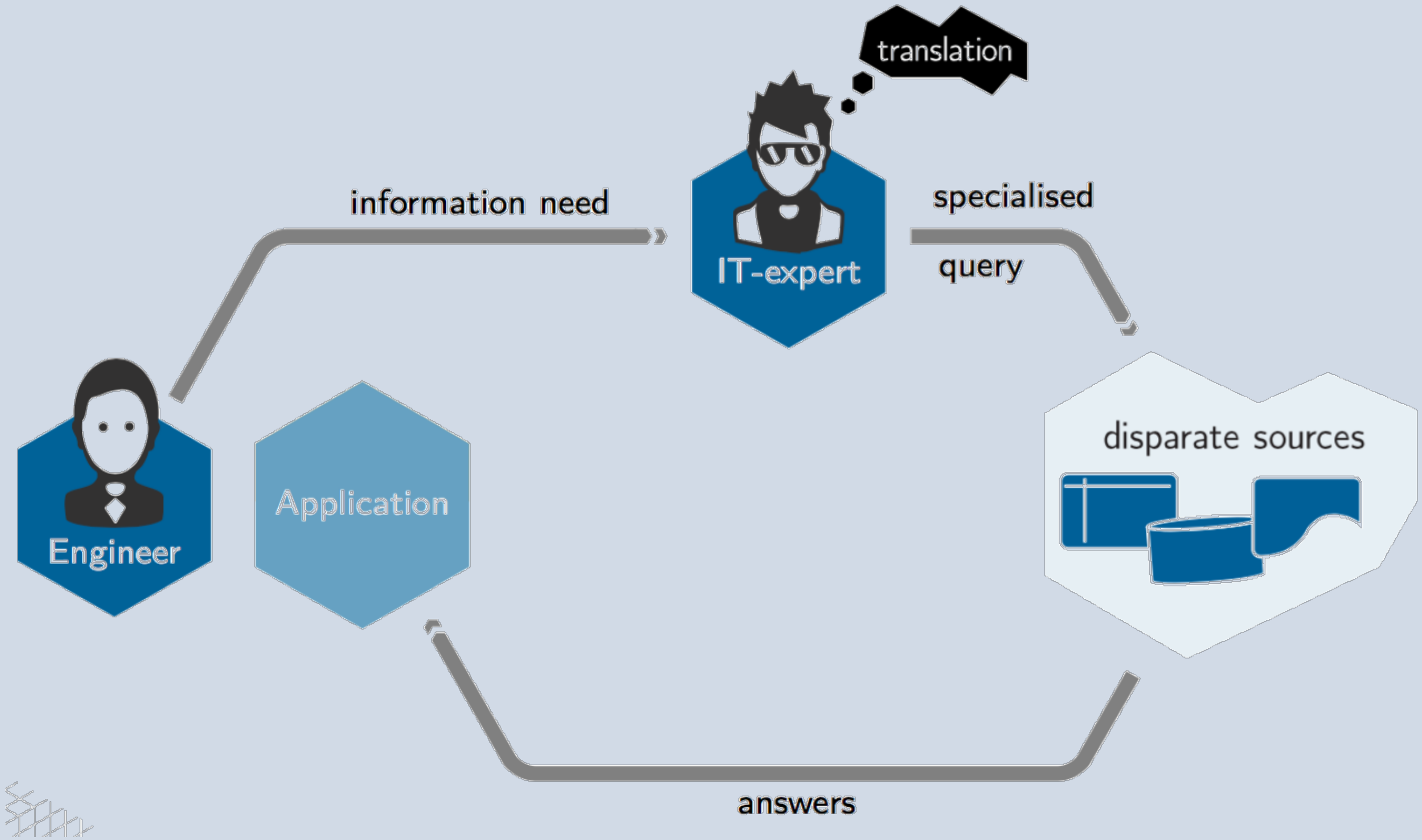
⋮

Data Access: Statoil Exploration



Information Systems Group

Data Access: Statoil Exploration



Information Systems Group

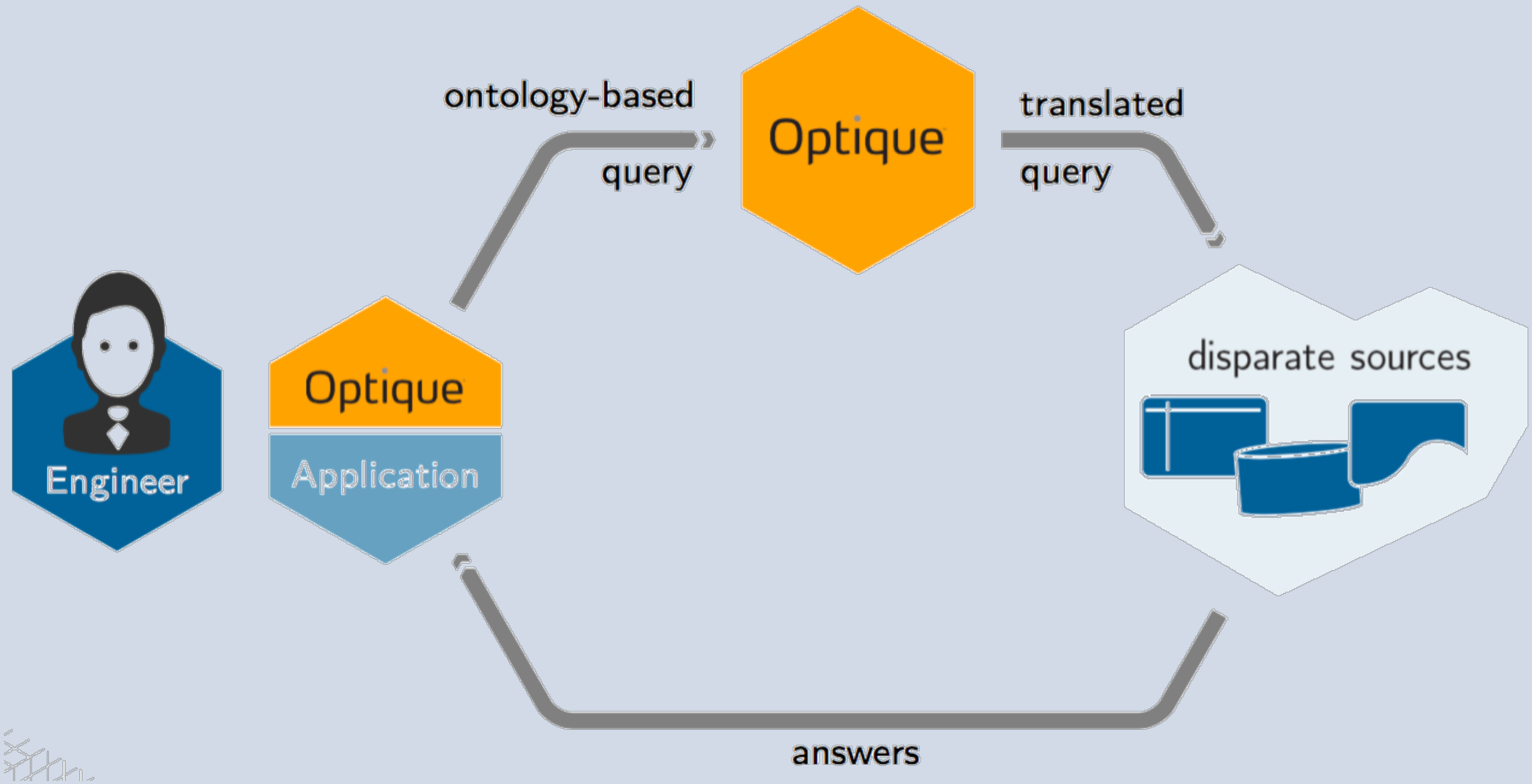
DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF OXFORD

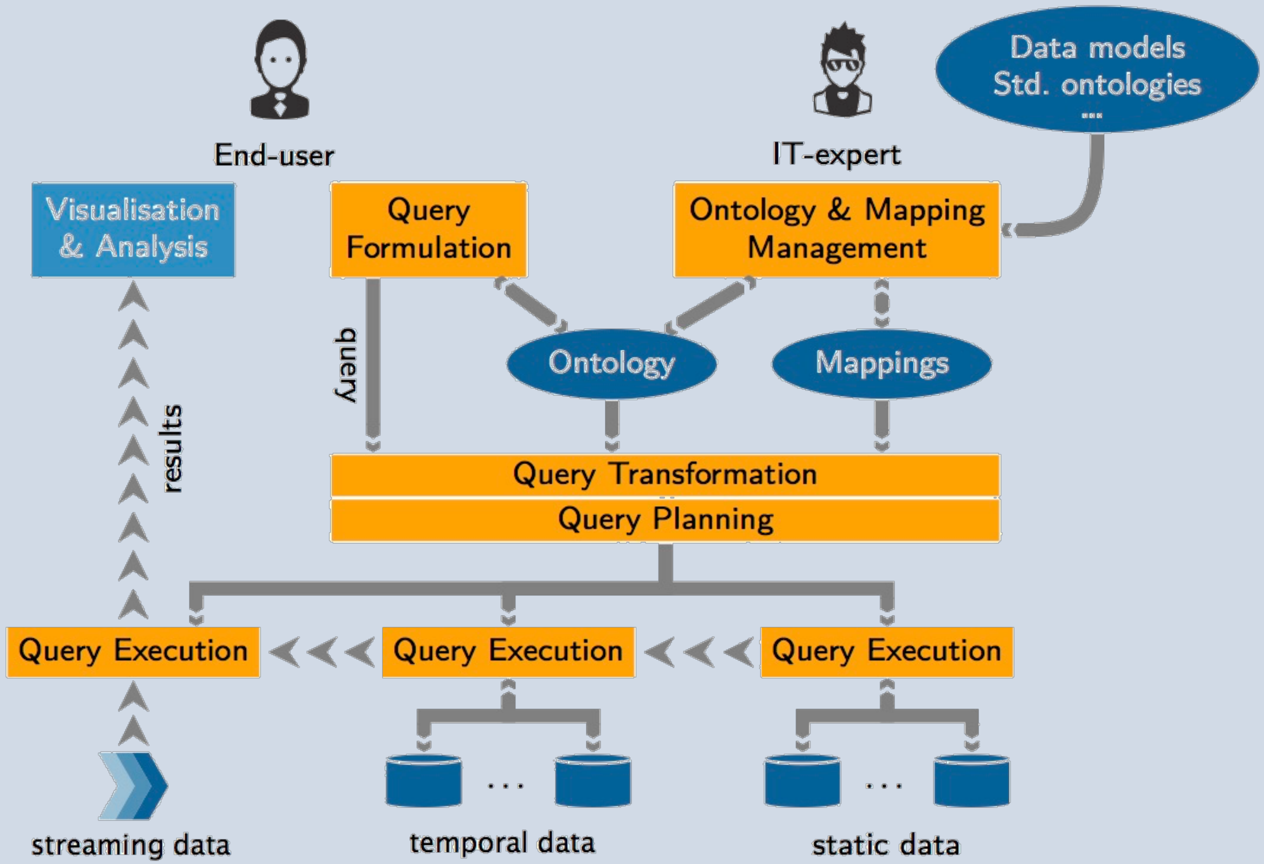
EPSRC Engineering and Physical Sciences Research Council

Optique

Data Access: Optique Approach

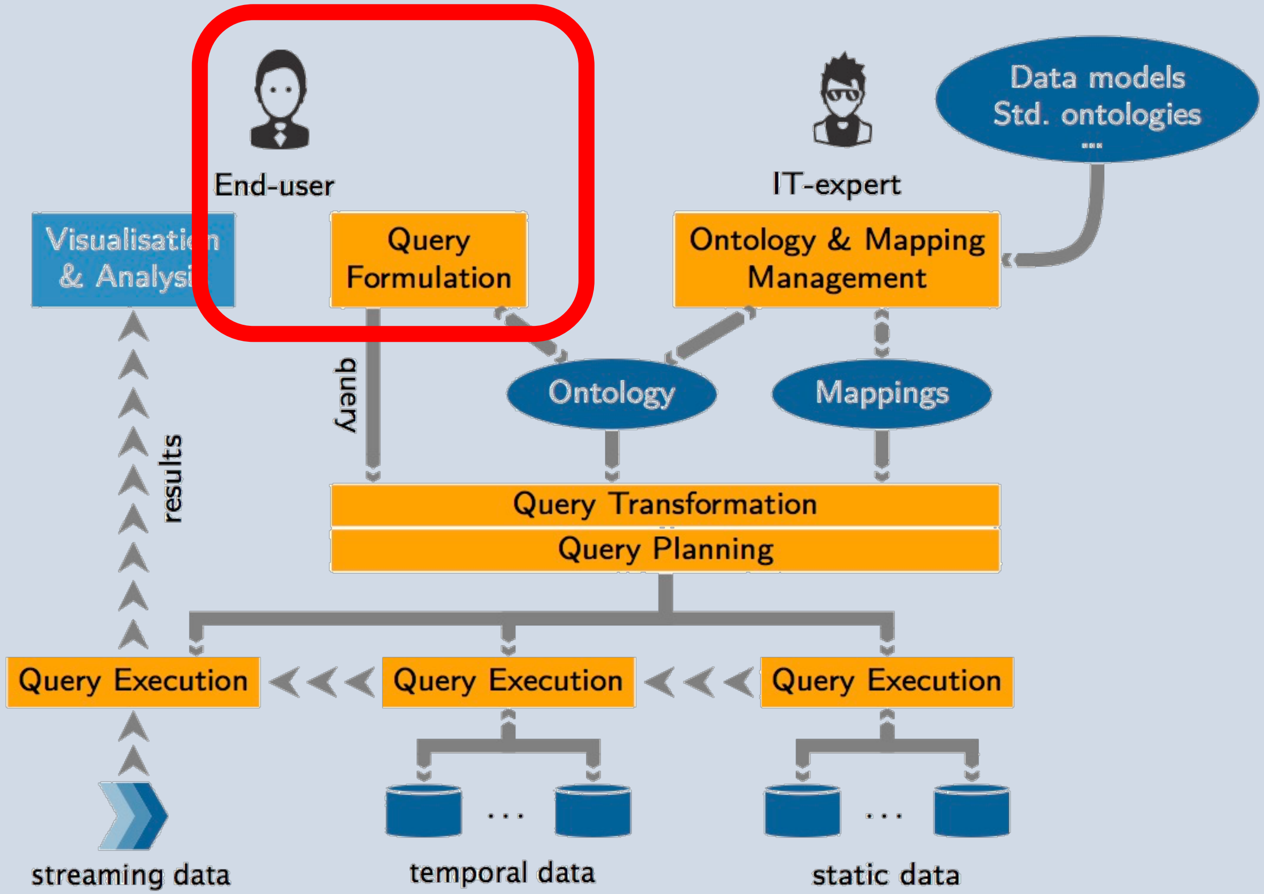


Data Access: Optique Approach

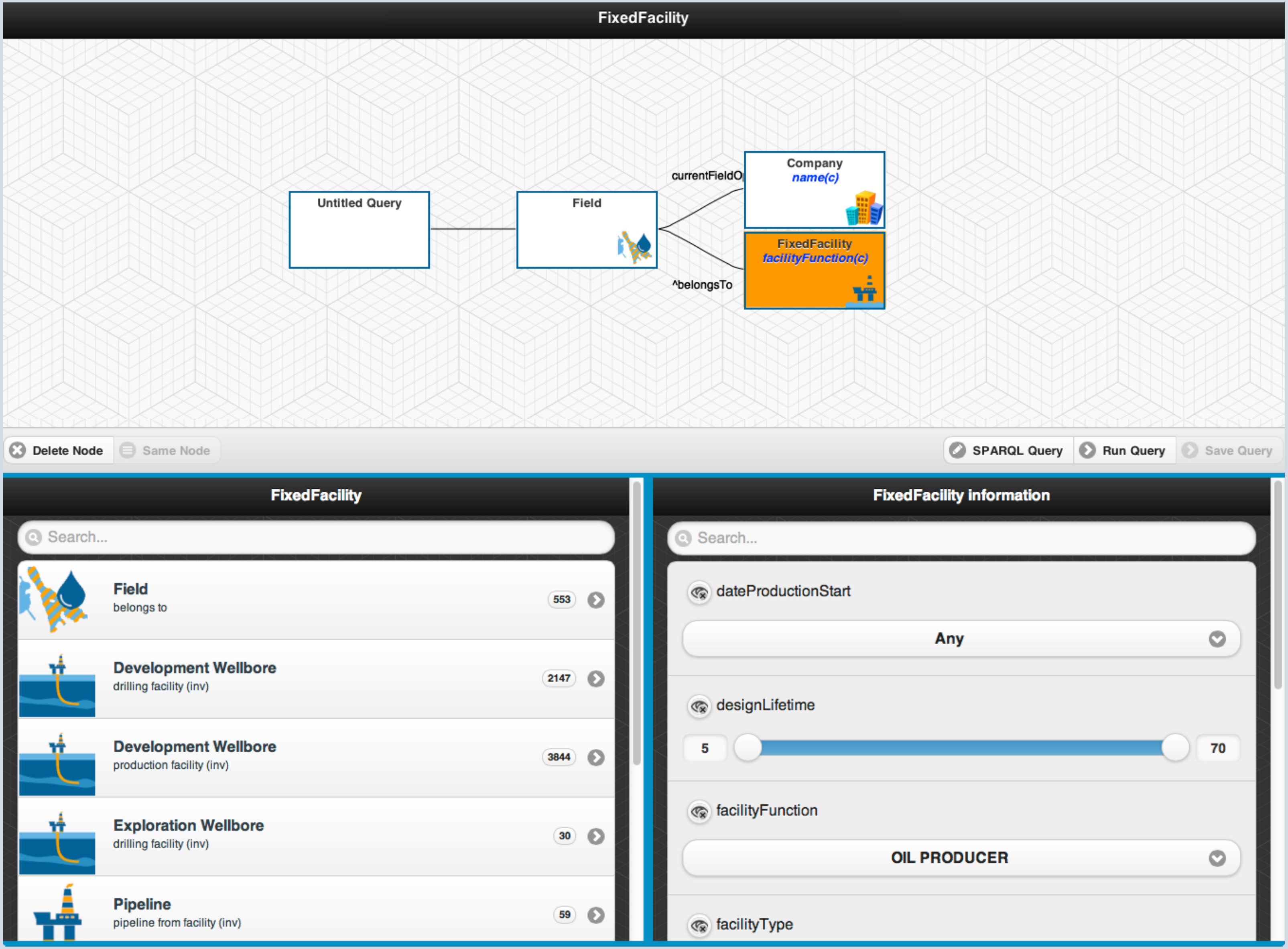


DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC
Engineering and Physical Sciences Research Council
Optique

Data Access: Optique Approach



UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique



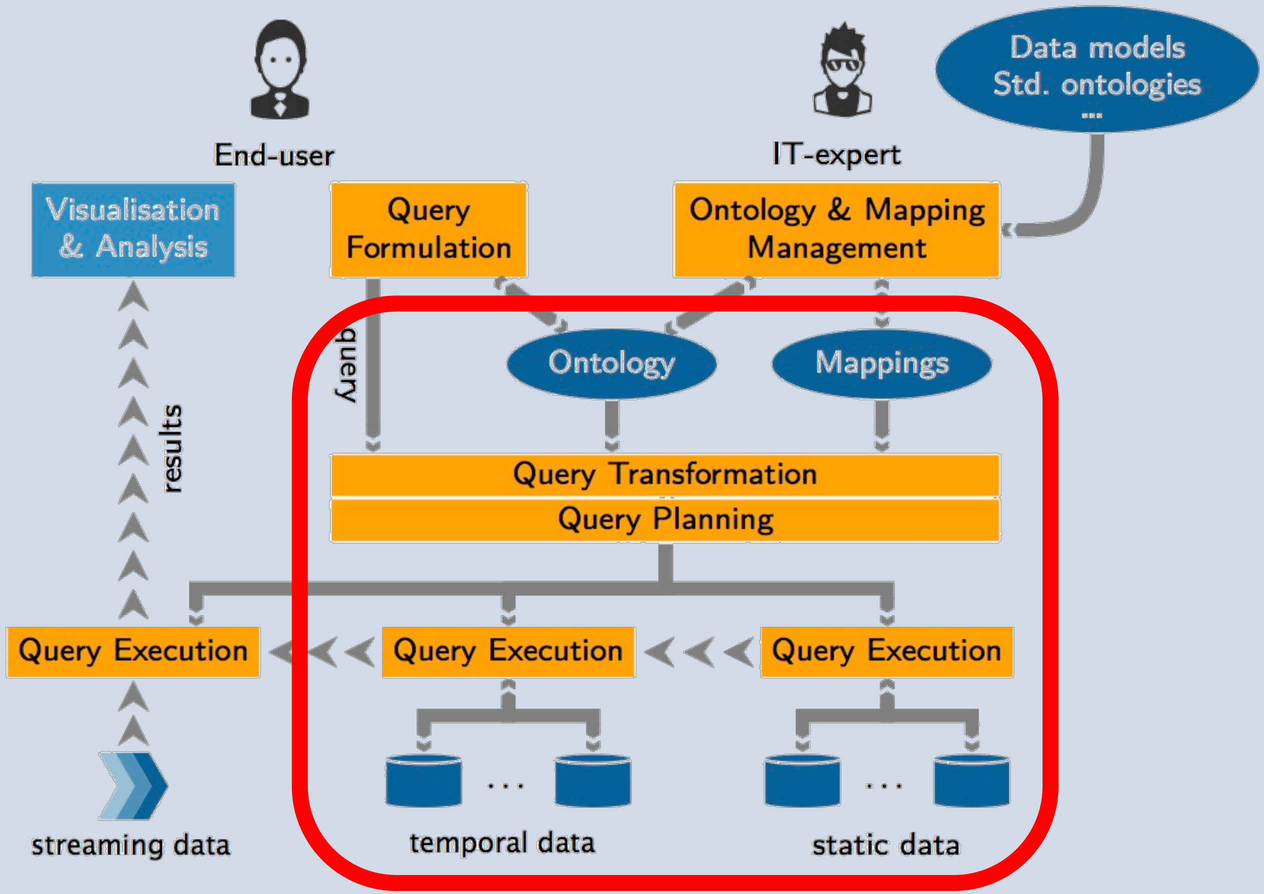
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD

Information Systems Group

EPSRC
Engineering and Physical Sciences Research Council

Optique

Data Access: Optique Approach



OWL Profiles

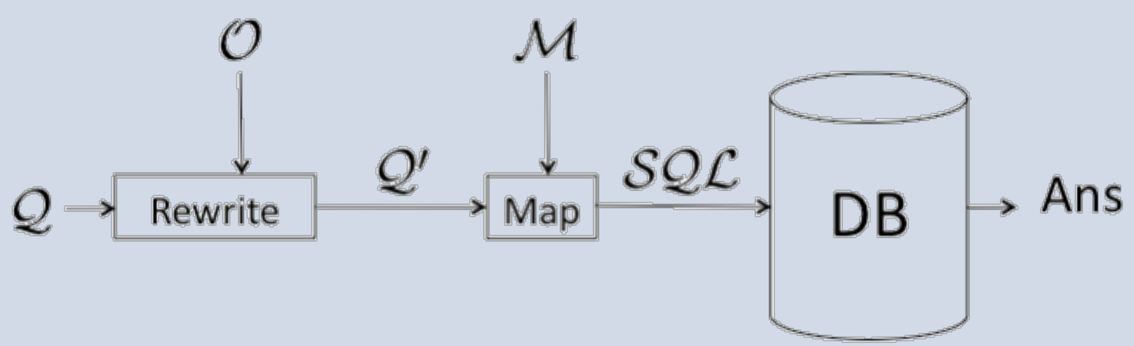
OWL 2 (2009) defines language subsets, aka **profiles** that can be "more simply and/or efficiently implemented"

- **OWL 2 QL**
  - Based on DL-Lite
  - Efficiently implementable via rewriting into relational queries (OBDA)

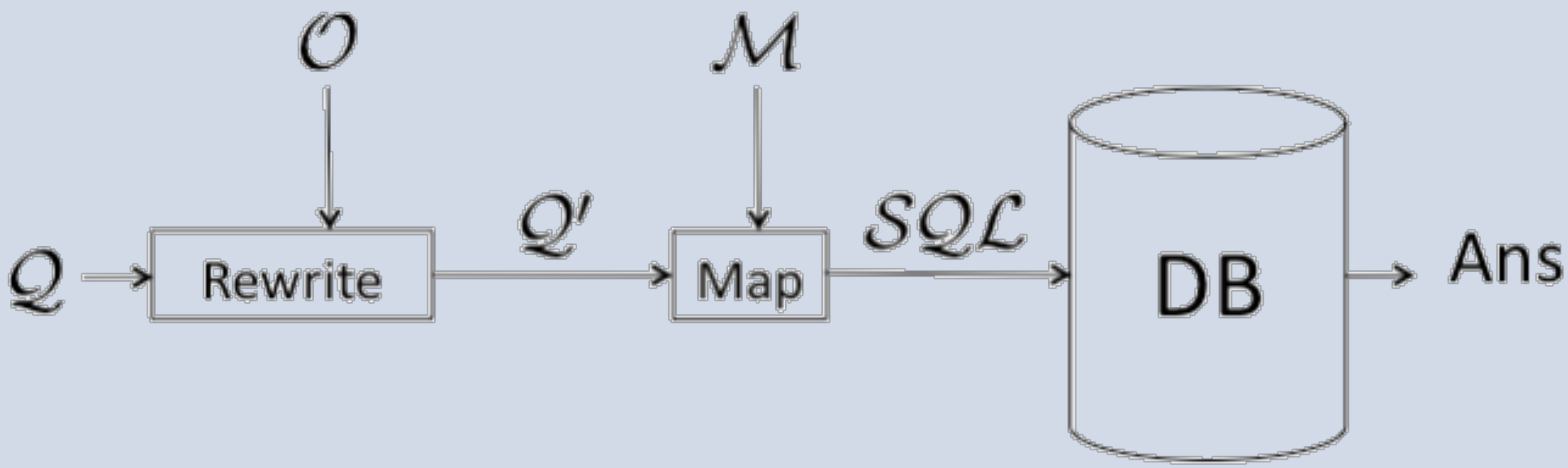
OWL 2 QL and Query Rewriting

Given QL ontology $\mathcal{O}$ query $\mathcal{Q}$ and mappings $\mathcal{M}$:

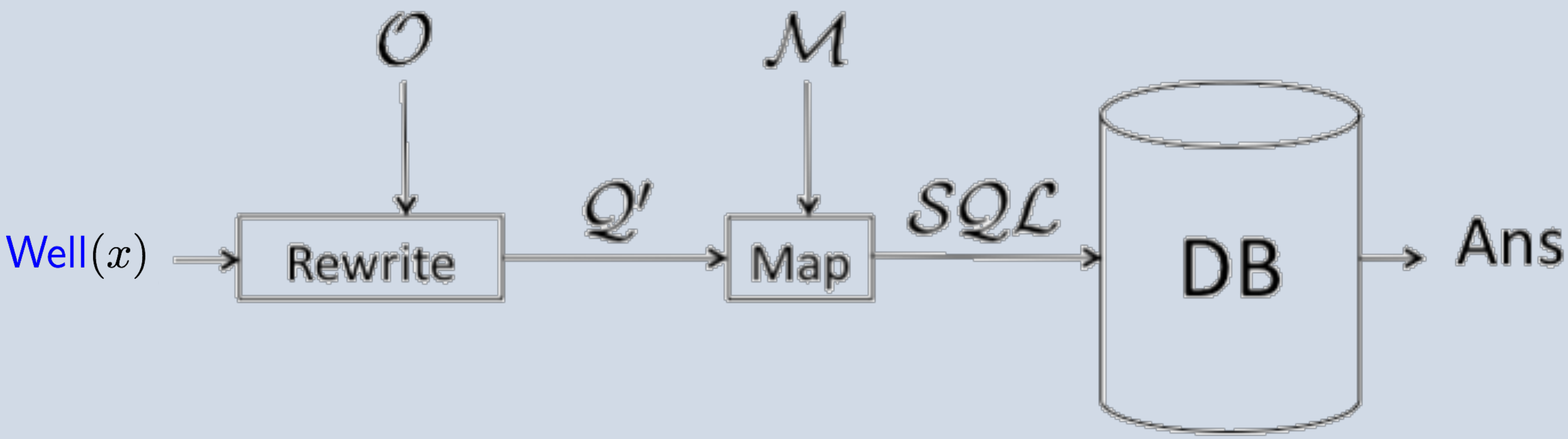
- Use $\mathcal{O}$ to **rewrite** $\mathcal{Q} \rightarrow \mathcal{Q}'$ s.t. answering $\mathcal{Q}'$ without $\mathcal{O}$ is equivalent to answering $\mathcal{Q}$ w.r.t. $\mathcal{O}$ *for any dataset*
- **Map** ontology queries $\rightarrow$ DB queries (typically SQL) using mappings $\mathcal{M}$ to rewrite $\mathcal{Q}'$ into a DB query
- **Evaluate** (SQL) query against DB



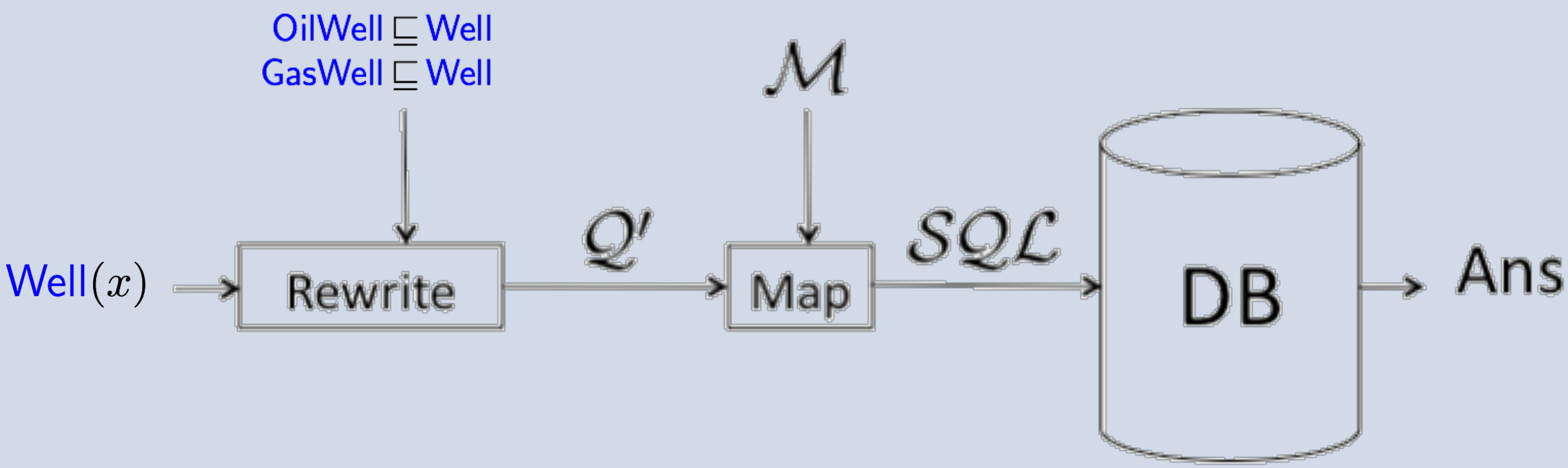
OWL 2 QL and Query Rewriting



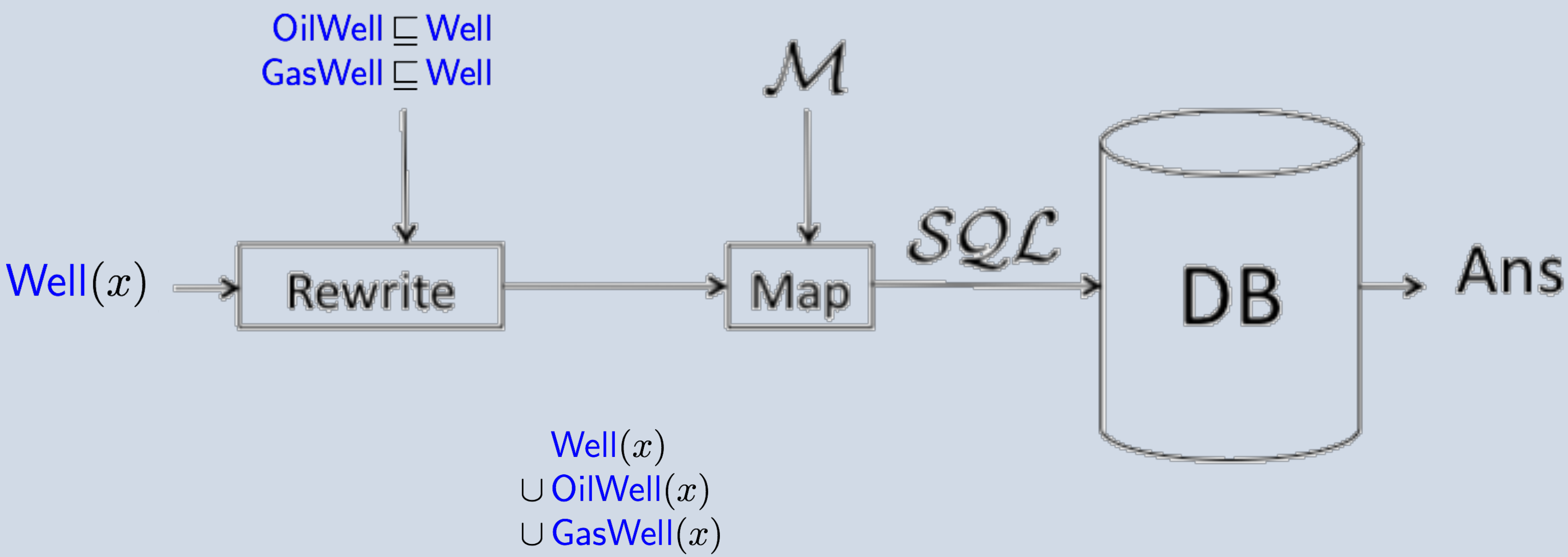
OWL 2 QL and Query Rewriting



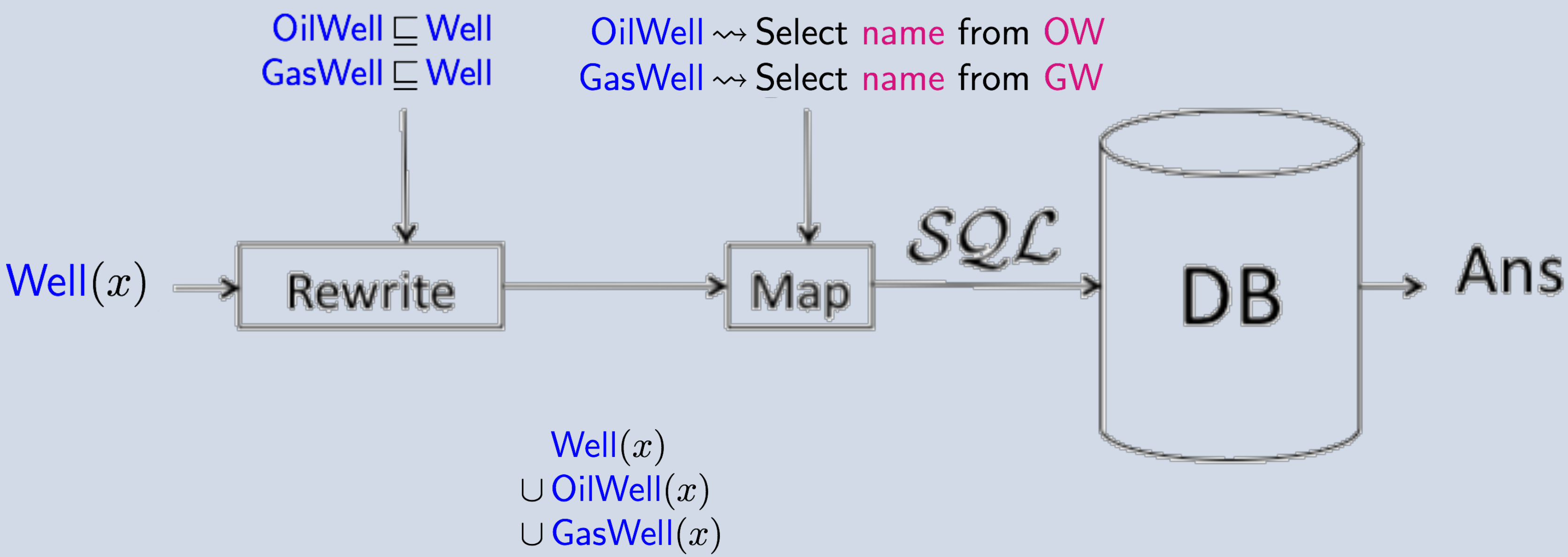
OWL 2 QL and Query Rewriting



OWL 2 QL and Query Rewriting



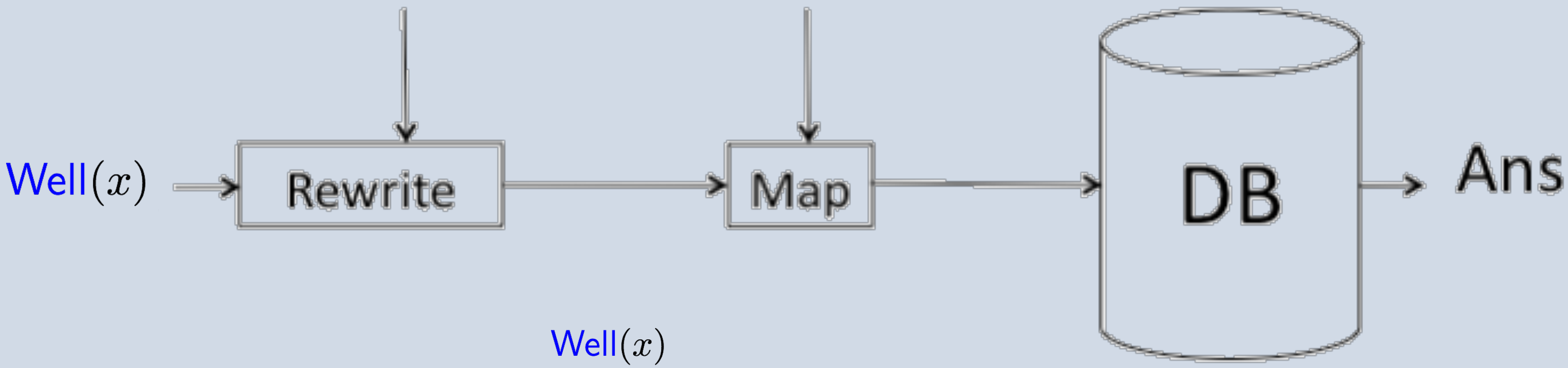
OWL 2 QL and Query Rewriting



OWL 2 QL and Query Rewriting

$\text{OilWell} \sqsubseteq \text{Well}$
$\text{GasWell} \sqsubseteq \text{Well}$

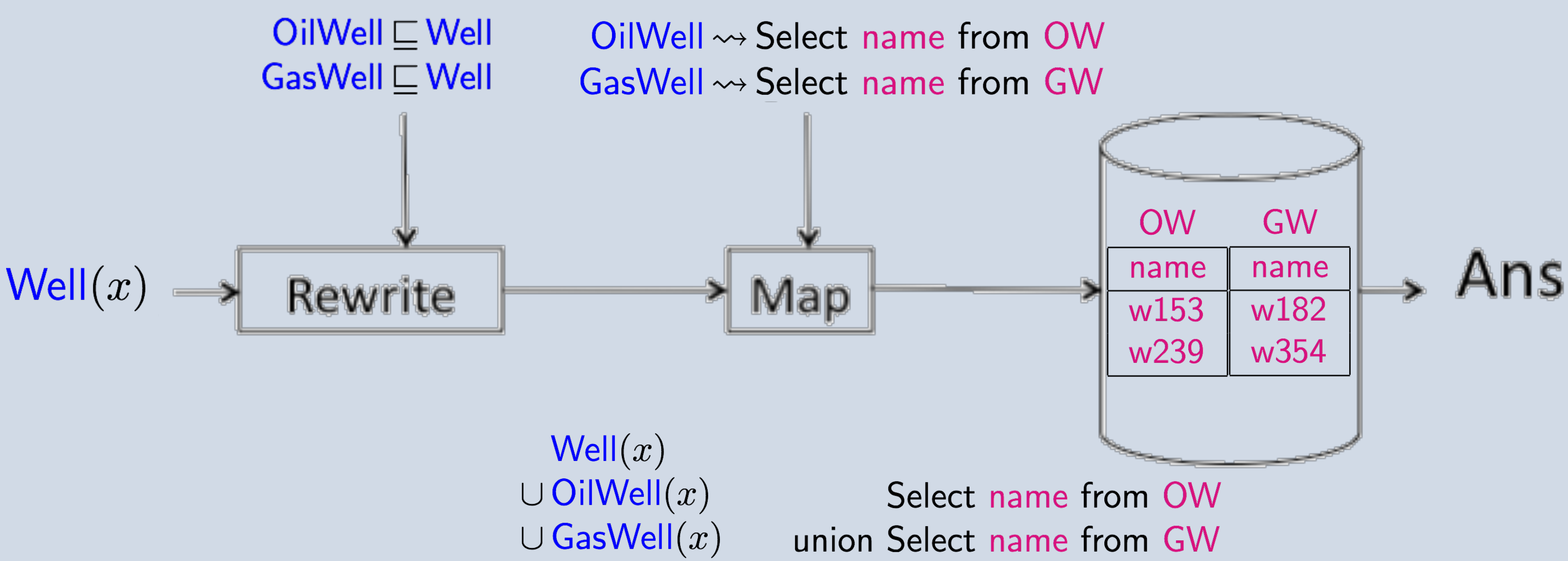
$\text{OilWell} \rightsquigarrow$ Select name from OW
$\text{GasWell} \rightsquigarrow$ Select name from GW



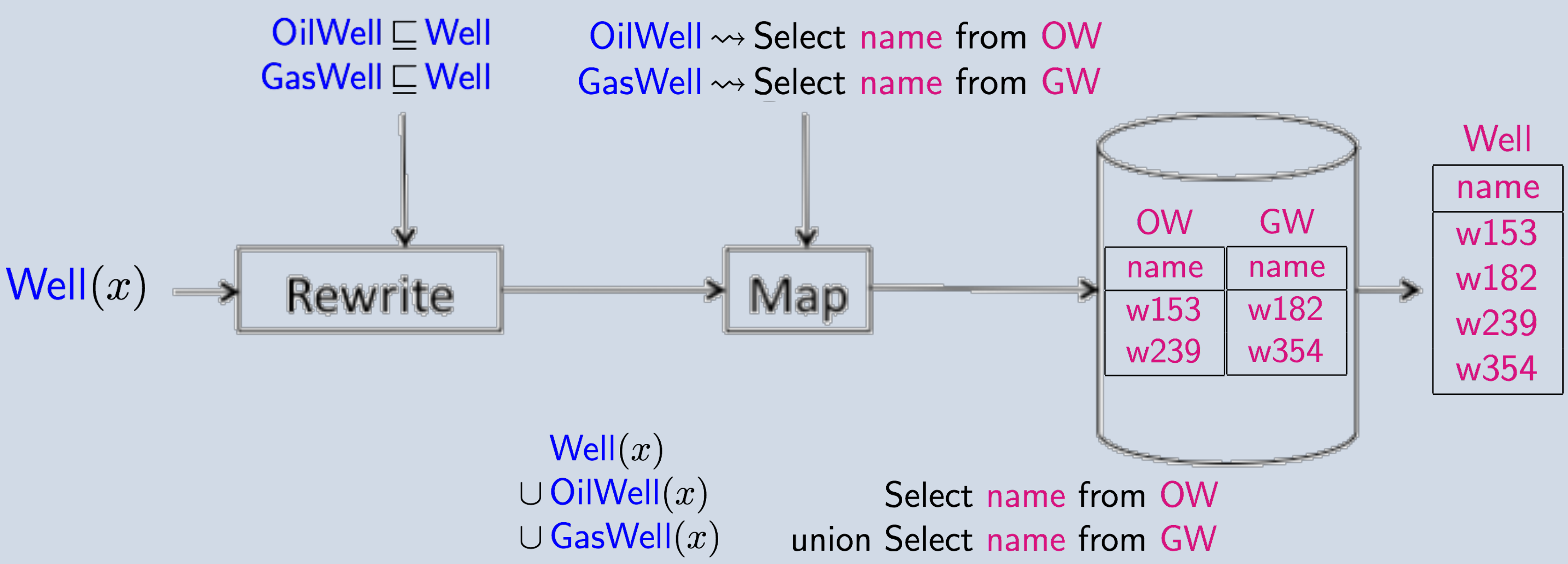
$\text{Well}(x)$
$\cup\, \text{OilWell}(x)$
$\cup\, \text{GasWell}(x)$

Select name from OW
union Select name from GW

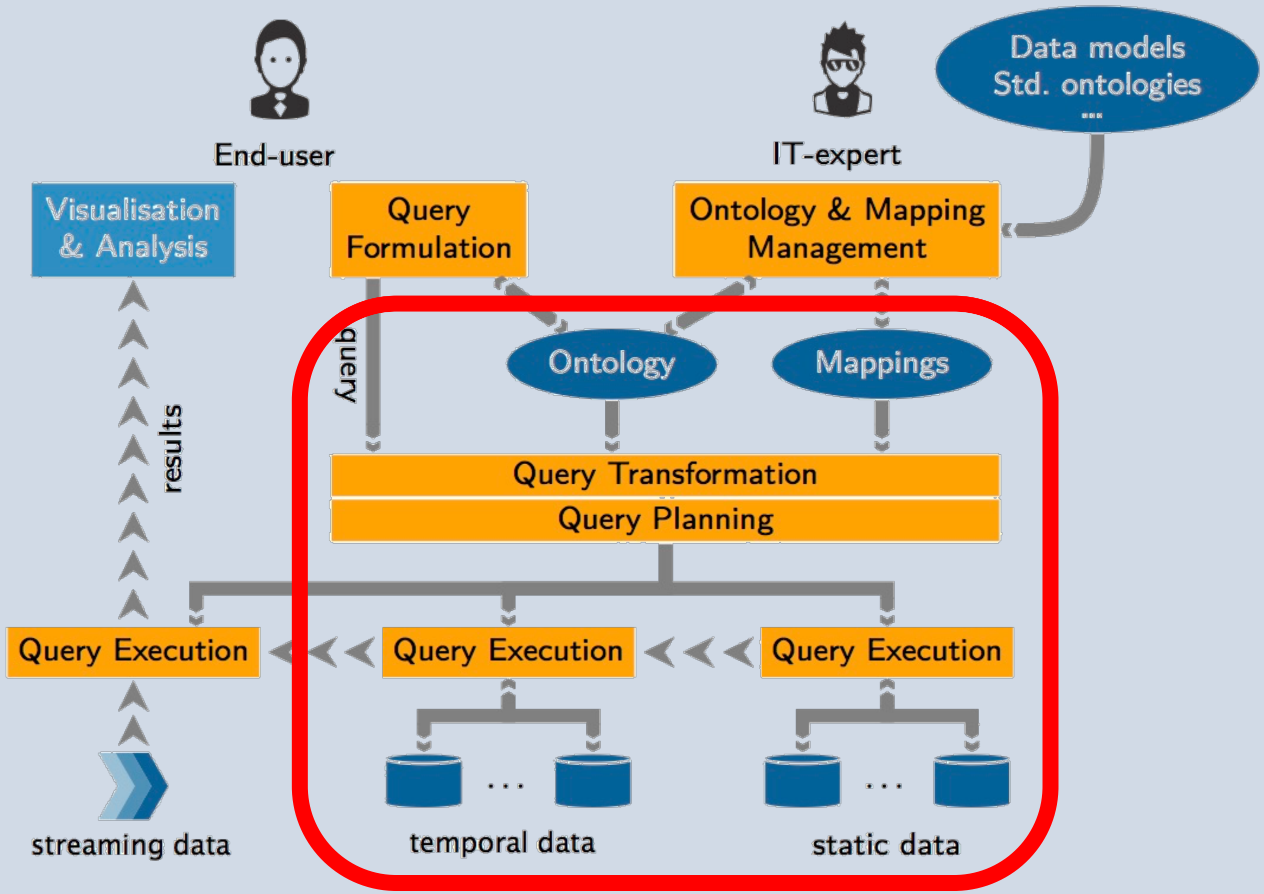
OWL 2 QL and Query Rewriting



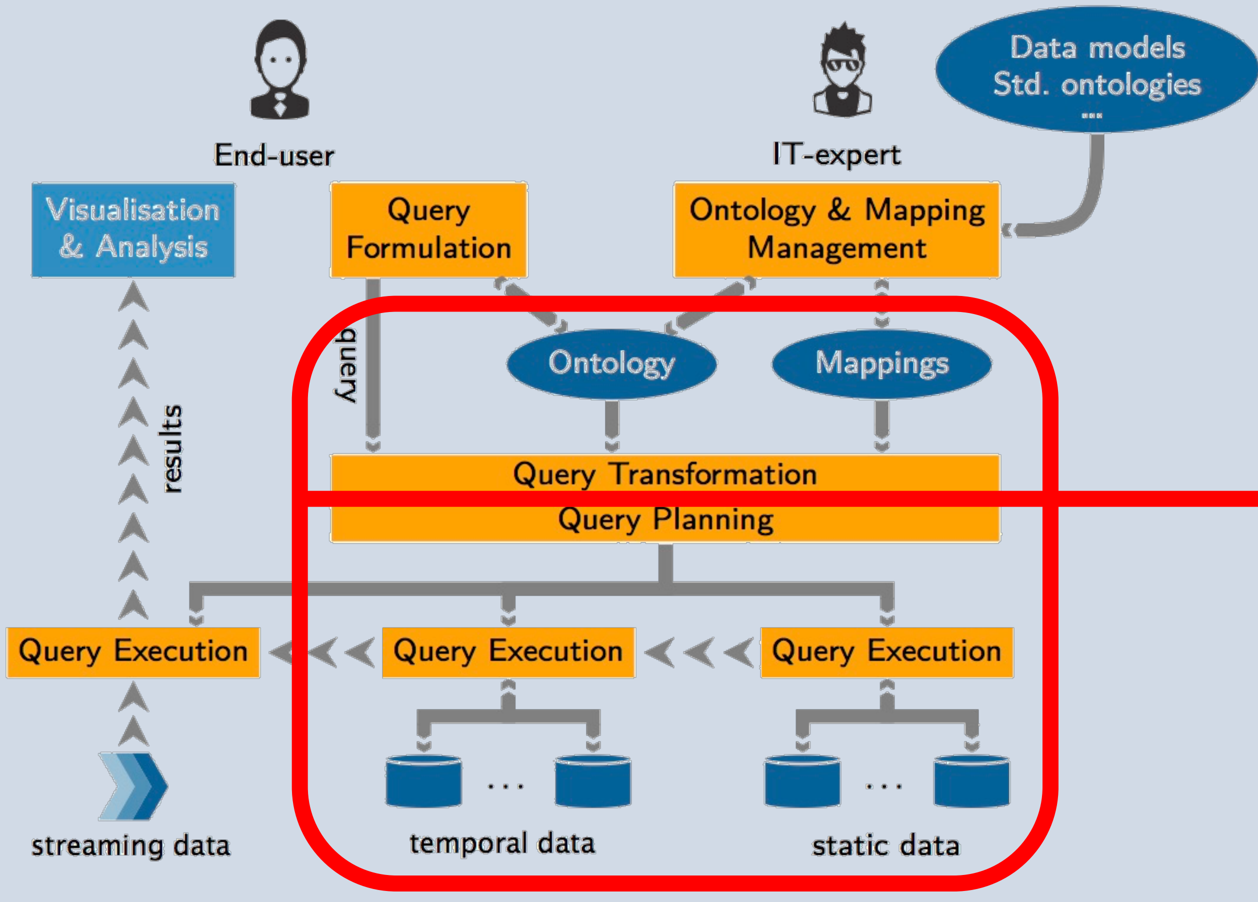
OWL 2 QL and Query Rewriting



Data Access: Optique Approach



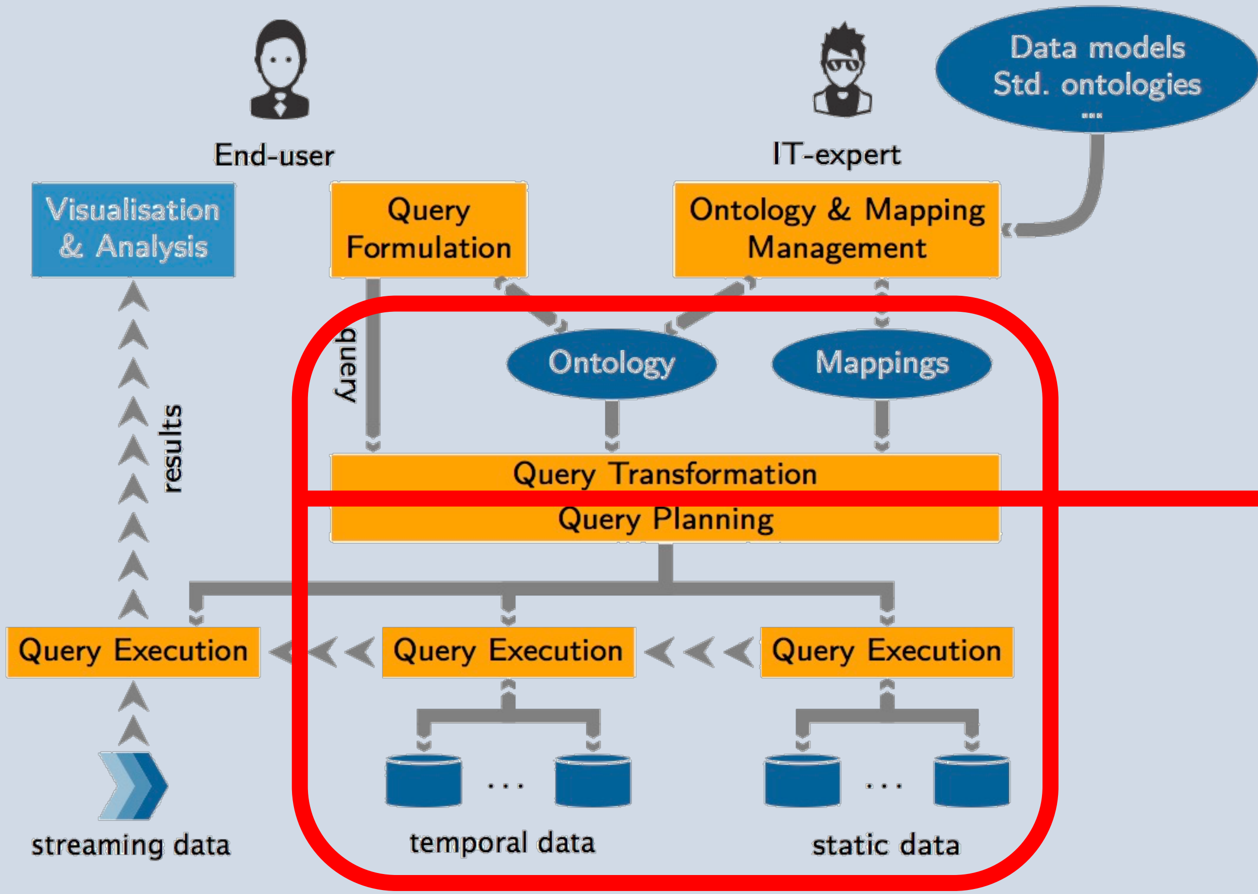
Data Access: Optique Approach



Query rewriting

- uses ontology & mappings
- computationally hard
- ontology & mappings small

Data Access: Optique Approach



Query rewriting

- uses ontology & mappings
- computationally hard
- ontology & mappings small

Query evaluation

- ind. of ontology & mappings
- computationally tractable
- data sets very large

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique

Data Access: Optique Approach

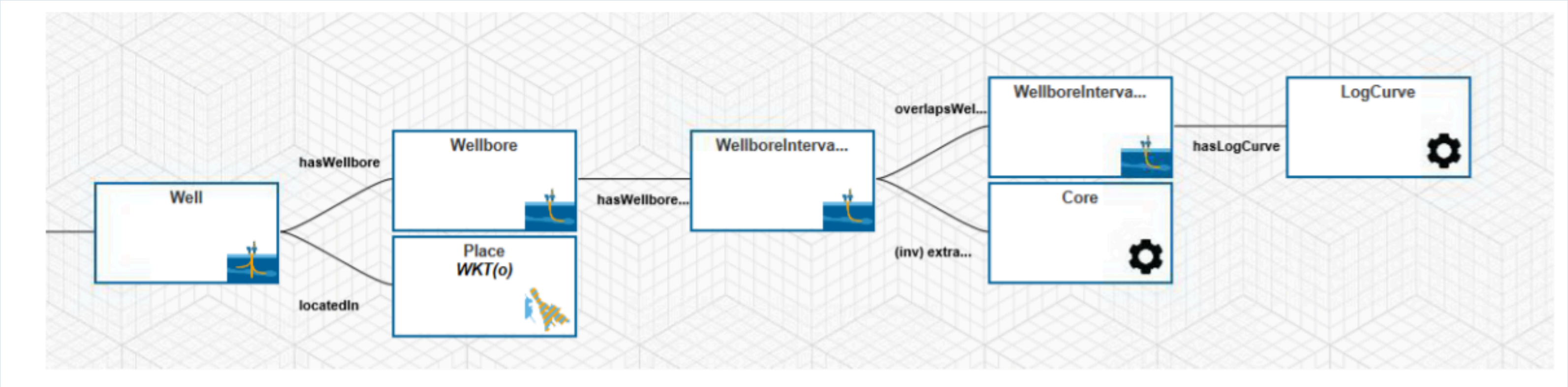
Problem Solved ?

Query Rewriting — Issues

1 Rewriting

- May be large (worst case exponential in size of ontology)
- Queries may be hard for existing DBMSs

Query: Wellbores with cores that overlap log curves



Query: Wellbores with cores that overlap log curves

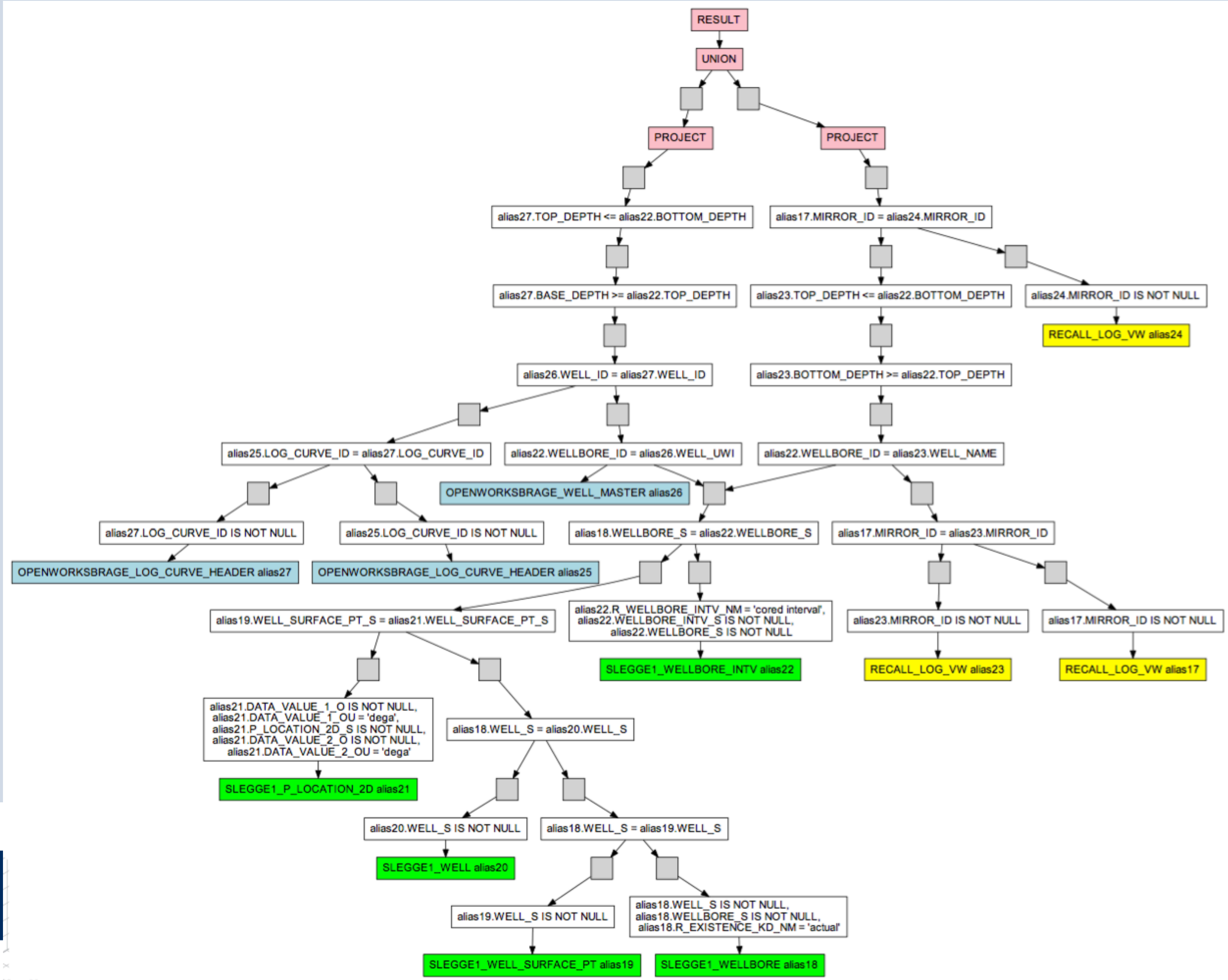
```
PREFIX ns1: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX ns2: <http://www.optique-project.eu/ontology/subsurface-exploration/>

SELECT DISTINCT ?Wellbore_c2 ?WKT_a1
WHERE {
  ?Well_c1 ns1:type ns2:Well.
  ?Wellbore_c2 ns1:type ns2:Wellbore.
  ?Place_c7 ns1:type ns2:Place.
  ?WellboreInterval_c3 ns1:type ns2:WellboreInterval.
  ?WellboreInterval_c4 ns1:type ns2:WellboreInterval.
  ?Core_c6 ns1:type ns2:Core.
  ?LogCurve_c5 ns1:type ns2:LogCurve.

  ?Well_c1 ns2:hasWellbore ?Wellbore_c2.
  ?Well_c1 ns2:locatedIn ?Place_c7.
  ?Place_c7 ns2:WKT ?WKT_a1.
  ?Wellbore_c2 ns2:hasWellboreInterval ?WellboreInterval_c3.
  ?WellboreInterval_c3 ns2:overlapsWellboreInterval ?WellboreInterval_c4.
  ?WellboreInterval_c3 ^ns2:extractedFrom ?Core_c6.
  ?WellboreInterval_c4 ns2:hasLogCurve ?LogCurve_c5.
}
```

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC
Engineering and Physical Sciences Research Council
Optique

Query: Wellbores with cores that overlap log curves

```
SELECT * FROM (

SELECT
 1 AS "Wellbore_c2QuestType", NULL AS "Wellbore_c2Lang",
 ('http://.../subsurface-exploration/Wellbore-' || QVIEW2.WELLBORE_S) AS "Wellbore_c2",
 3 AS "WKT_a1QuestType", NULL AS "WKT_a1Lang",
 (((('POINT(' || QVIEW5.DATA_VALUE_2_O) || ' ') || QVIEW5.DATA_VALUE_1_O) || ')') AS "WKT_a1"

FROM ADP.SLEGGE1_WELLBORE QVIEW2,
     ADP.SLEGGE1_WELL_SURFACE_PT QVIEW3,
     ADP.SLEGGE1_WELL QVIEW4,
     ADP.SLEGGE1_P_LOCATION_2D QVIEW5,
     ADP.SLEGGE1_WELLBORE_INTV QVIEW6,
     ADP.RECALL_LOG_VW QVIEW1,
     ADP.RECALL_LOG_VW QVIEW7,
     ADP.RECALL_LOG_VW QVIEW8
WHERE QVIEW1.MIRROR_ID IS NOT NULL
  AND (QVIEW2.R_EXISTENCE_KD_NM = 'actual')
  AND QVIEW2.WELL_S IS NOT NULL
  AND QVIEW2.WELLBORE_S IS NOT NULL
  AND (QVIEW2.WELL_S = QVIEW3.WELL_S)
  AND (QVIEW2.WELL_S = QVIEW4.WELL_S)
  AND (QVIEW5.DATA_VALUE_1_OU = 'dega')
  AND (QVIEW5.DATA_VALUE_2_OU = 'dega')
  AND (QVIEW3.WELL_SURFACE_PT_S = QVIEW5.WELL_SURFACE_PT_S)
  AND QVIEW5.P_LOCATION_2D_S IS NOT NULL
  AND (QVIEW2.WELLBORE_S = QVIEW6.WELLBORE_S)
  AND (QVIEW6.R_WELLBORE_INTV_NM = 'cored interval')
  AND QVIEW6.WELLBORE_INTV_S IS NOT NULL
  AND (QVIEW1.MIRROR_ID = QVIEW7.MIRROR_ID)
  AND (QVIEW6.WELLBORE_ID = QVIEW7.WELL_NAME)
  AND ((QVIEW7.BOTTOM_DEPTH >= QVIEW6.TOP_DEPTH)
       AND (QVIEW7.TOP_DEPTH <= QVIEW6.BOTTOM_DEPTH))
  AND (QVIEW1.MIRROR_ID = QVIEW8.MIRROR_ID)
  AND QVIEW5.DATA_VALUE_1_O IS NOT NULL
  AND QVIEW5.DATA_VALUE_2_O IS NOT NULL

UNION
SELECT ...

FROM ADP.SLEGGE1_WELLBORE QVIEW2,
     ADP.SLEGGE1_WELL_SURFACE_PT QVIEW3,
     ADP.SLEGGE1_WELL QVIEW4,
     ADP.SLEGGE1_P_LOCATION_2D QVIEW5,
     ADP.SLEGGE1_WELLBORE_INTV QVIEW6,
     ADP.OPENWORKSBRAGE_LOG_CURVE_HEADER QVIEW1,
     ADP.OPENWORKSBRAGE_WELL_MASTER QVIEW7,
     ADP.OPENWORKSBRAGE_LOG_CURVE_HEADER QVIEW8
WHERE  QVIEW1.LOG_CURVE_ID IS NOT NULL
  AND  (QVIEW2.R_EXISTENCE_KD_NM = 'actual')
  AND  QVIEW2.WELL_S IS NOT NULL
  AND  QVIEW2.WELLBORE_S IS NOT NULL
  AND  (QVIEW2.WELL_S = QVIEW3.WELL_S)
  AND  (QVIEW2.WELL_S = QVIEW4.WELL_S)
  AND  (QVIEW5.DATA_VALUE_1_OU = 'dega')
  AND  (QVIEW5.DATA_VALUE_2_OU = 'dega')
  AND  (QVIEW3.WELL_SURFACE_PT_S = QVIEW5.WELL_SURFA(
  AND  QVIEW5.P_LOCATION_2D_S IS NOT NULL
  AND  (QVIEW2.WELLBORE_S = QVIEW6.WELLBORE_S)
  AND  (QVIEW6.R_WELLBORE_INTV_NM = 'cored interval')
  AND  QVIEW6.WELLBORE_INTV_S IS NOT NULL
  AND  (QVIEW6.WELLBORE_ID = QVIEW7.WELL_UWI)
  AND  (QVIEW1.LOG_CURVE_ID = QVIEW8.LOG_CURVE_ID)
  AND  (QVIEW7.WELL_ID = QVIEW8.WELL_ID)
  AND  ((QVIEW8.BASE_DEPTH >= QVIEW6.TOP_DEPTH)
         AND  (QVIEW8.TOP_DEPTH <= QVIEW6.BOTTOM_DEPTI
  AND  QVIEW5.DATA_VALUE_1_O IS NOT NULL
  AND  QVIEW5.DATA_VALUE_2_O IS NOT NULL)
SUB_QVIEW
```

Query: Wellbores with cores that overlap log curves



Query Rewriting — Issues

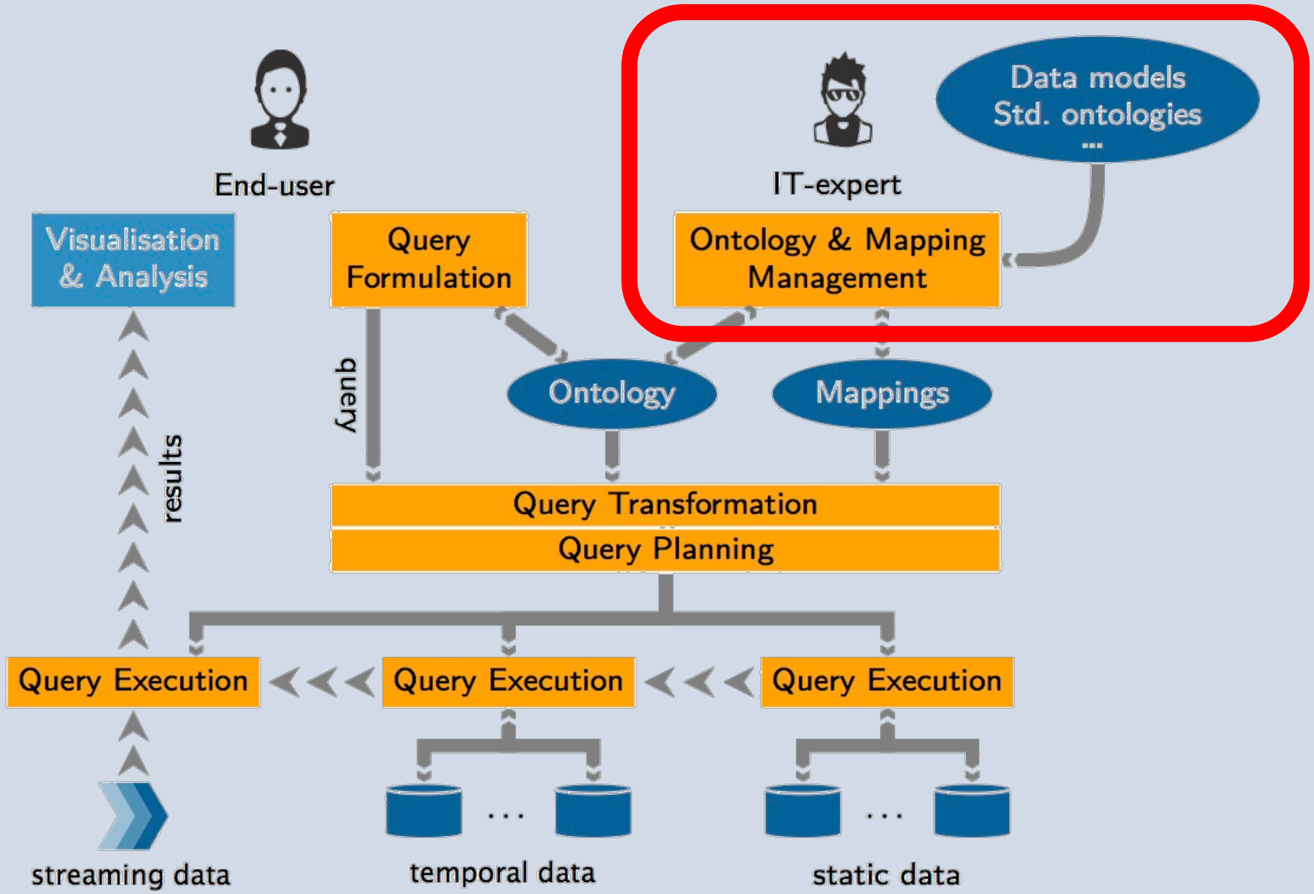
1 Rewriting

- May be large (worst case exponential in size of ontology)
- Queries may be hard for existing DBMSs

2 Ontology & Mappings

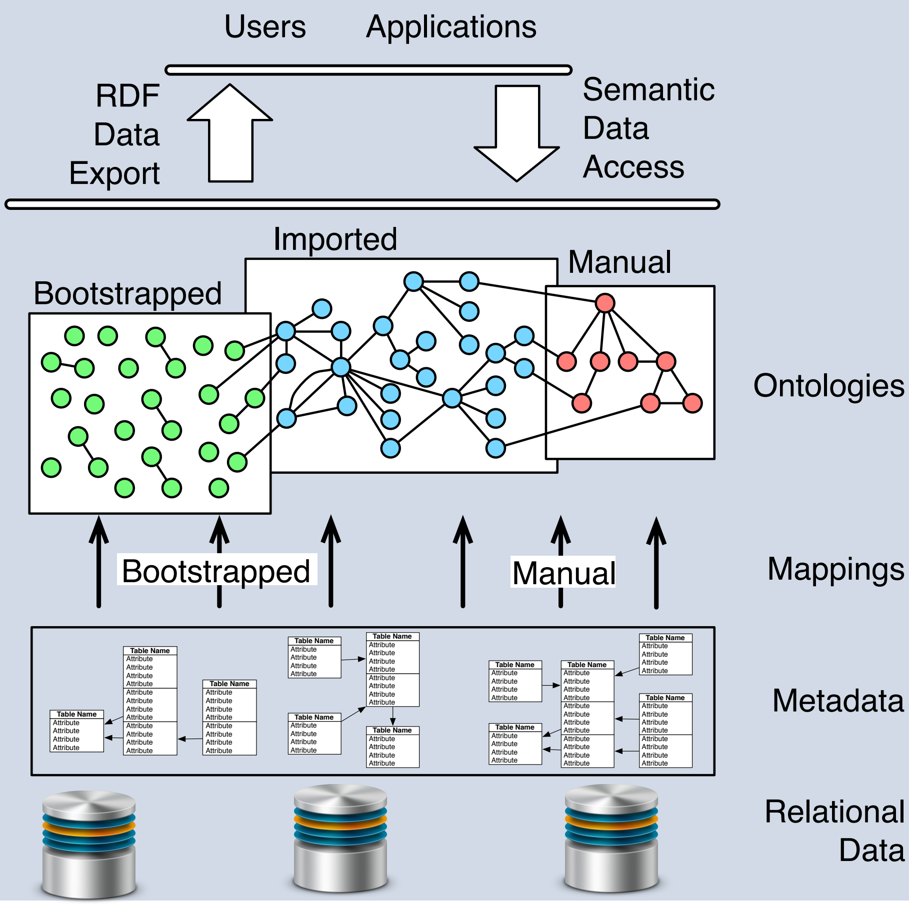
- May be difficult to develop and maintain

Data Access: Optique Approach



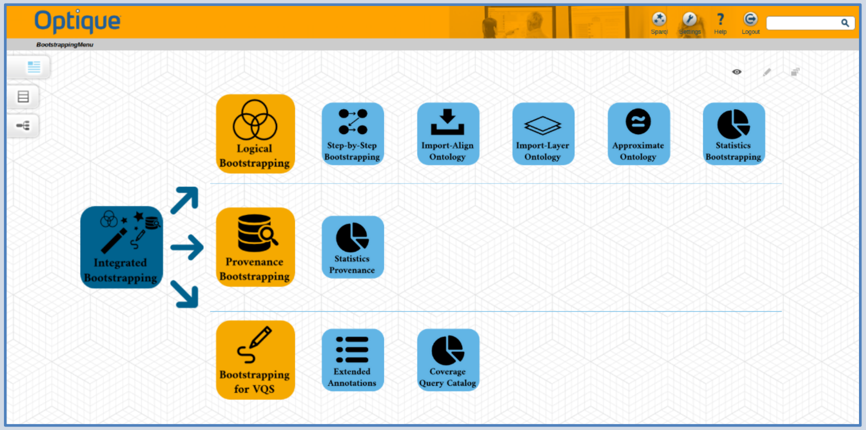
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD
Information Systems Group
EPSRC
Engineering and Physical Sciences Research Council
Optique

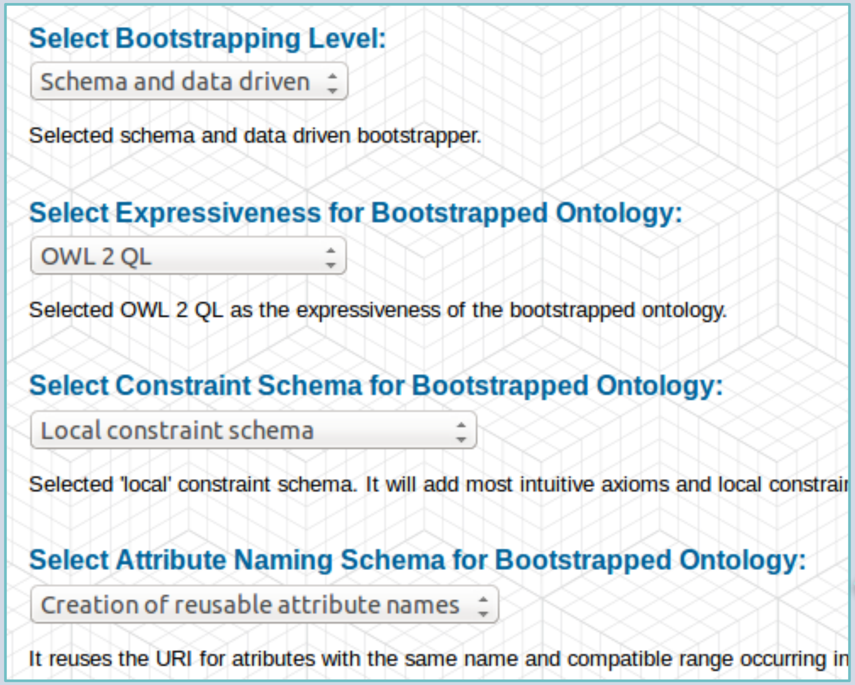
BootOX

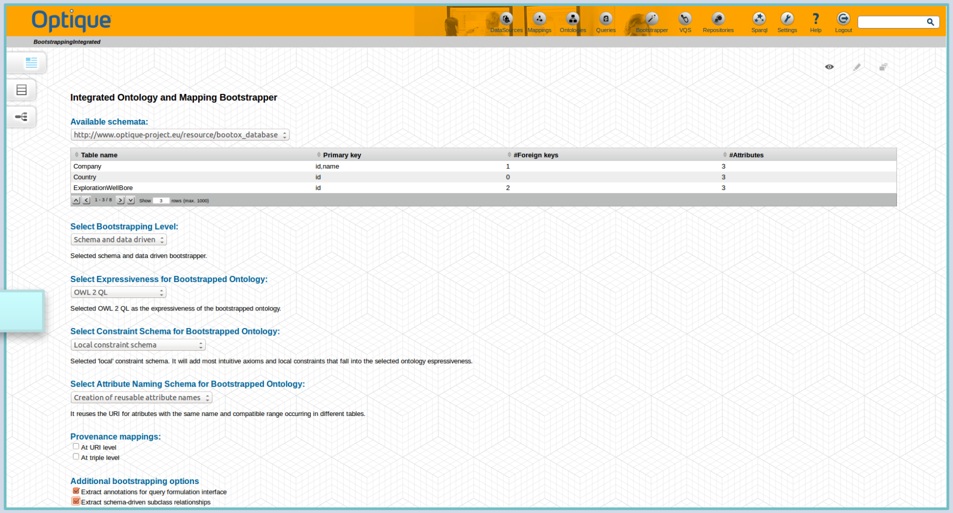


Users
Applications
RDF
Data
Export
Semantic
Data
Access
Imported
Manual
Bootstrapped
Ontologies
Bootstrapped
Manual
Mappings
Metadata
Relational
Data

BootOX







UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

EPSRC Engineering and Physical Sciences Research Council

Optique

Query Rewriting — Issues

1 Rewriting

- May be large (worst case exponential in size of ontology)
- Queries may be hard for existing DBMSs

2 Ontology & Mappings

- May be difficult to develop and maintain

3 Expressivity

- OWL 2 QL (necessarily) has (very) restricted expressive power, e.g.:
  - No functional or transitive properties
  - No universal (for-all) restrictions
  - …

OWL Profiles – Beyond QL?

OWL 2 (2009) defines language subsets, aka **profiles** that can be "more simply and/or efficiently implemented"

- **OWL 2 QL**
  - Based on DL-Lite
  - Efficiently implementable via rewriting into relational queries (OBDA)
- **OWL 2 RL**
  - Based on "**Description Logic Programs**" ($\approx$ DL $\cap$ Datalog)
  - Implementable via Datalog query answering
- **OWL 2 EL**
  - Based on $\mathcal{EL}^{++}$
  - Implementable via Datalog query answering plus "filtration"

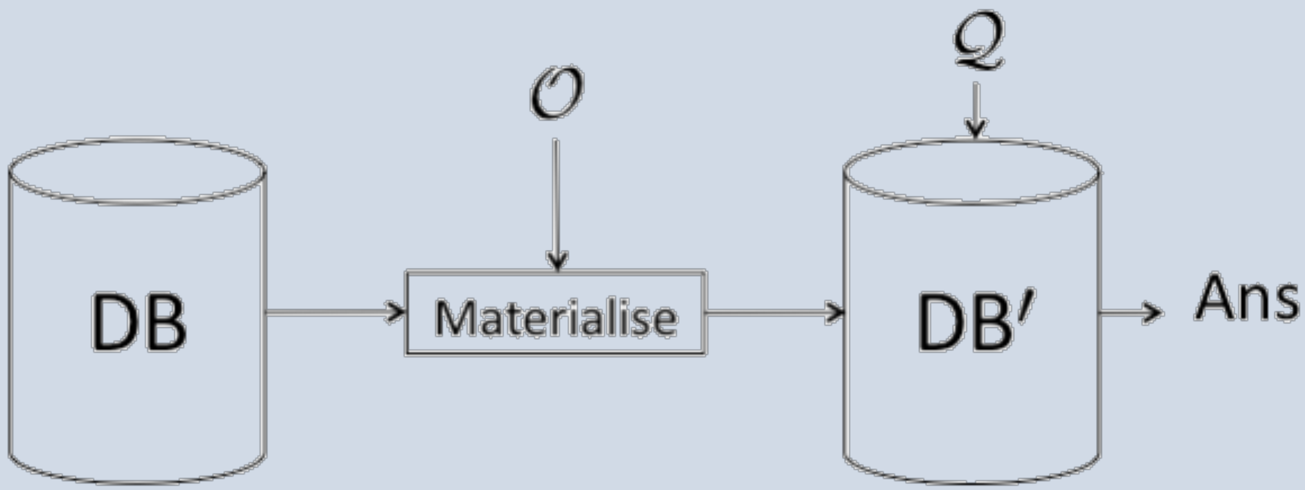
RL/Datalog Query Ans. via Materialisation

Given (RDF) data DB, RL/Datalog ontology $\mathcal{O}$ and query $\mathcal{Q}$:

- **Materialise** (RDF) data DB $\longrightarrow$ DB$'$ s.t. evaluating $\mathcal{Q}$ w.r.t. DB$'$ equivalent to answering $\mathcal{Q}$ w.r.t. DB and $\mathcal{O}$

  nb: Closely related to **chase** procedure used with DB dependencies

- **Evaluate** $\mathcal{Q}$ against DB$'$



Materialisation — Issues

1 Scalability

- Ptime complete
- Efficiently implementable in practice?

2 Updates

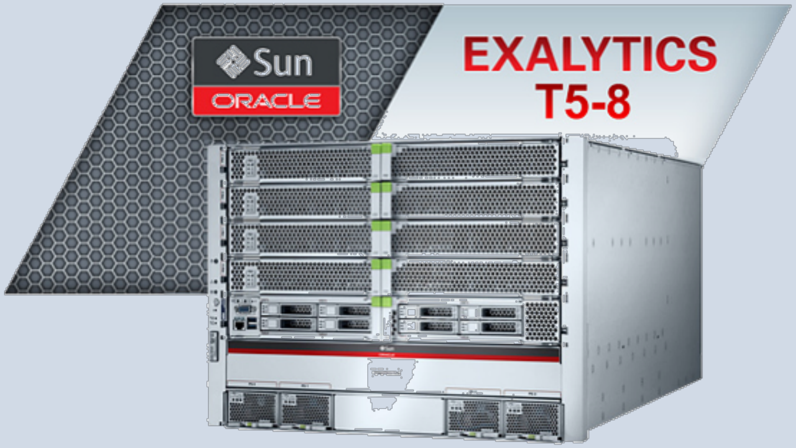
- Additions relatively easy (continue materialisation)
- But what about retraction?

Materialisation: Scalability

- Efficient **Datalog/RL** engine is critical
- Existing approaches mainly target distributed “shared-nothing” architectures, often via **map reduce**
  - High communication overhead
  - Typically focus on small fragments (e.g., RDFS), so don’t really address expressivity issue
  - Even then, query answering over (distributed) materialized data is non-trivial and may require considerable communication

RDFox Datalog/RL Engine

- Targets SOTA **main-memory, mulit-core** architecture
  - Optimized in-memory storage with ‘mostly’ lock-free parallel inserts
  - Memory efficient: commodity server with 128 GB can store $>10^9$ triples
  - Exploits multi-core architecture: 10-20 x speedup with 32/16 threads/cores
  - LUBM 120K ($>10^{10}$ triples) in 251s (20M t/s) on T5-8 (4TB/1024 threads)



RDFox Datalog/RL Engine

- **Incremental addition and retraction** of triples
  - Retraction via novel FBF “view maintenance” algorithm
  - Retraction of 5,000 triples from materialised LUBM 50k in less than 1s
- Many other **novel features**
  - Handles more general (than RL) Dalalog and SWRL rules
  - SPARQL features such as BIND and FILTER in rule bodies
  - Native equality handling (owl:sameAs) via rewriting
  - Stratified negation as failure (NAF)
  - …

Materialisation — Issues

1 Scalability

- Ptime complete
- Efficiently implementable in practice?

2 Updates

- Additions relatively easy (continue materialisation)
- But what about retraction?

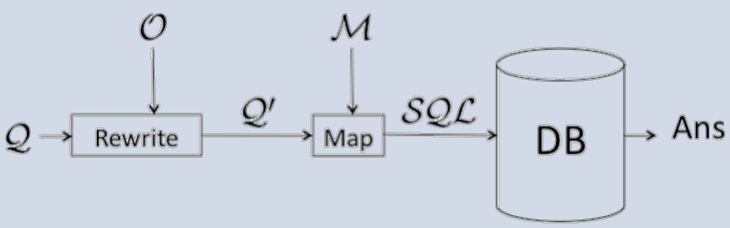
3 Migrating data to RDF

- Materialisation assumes data in “special” (RDF triple) store
- How can legacy data be migrated?

Materialisation: Data Migration

- Need to specify a suitable **migration** process
  - Use **R2RML** mappings to extract data and transform into RDF
  - But where do these mappings come from?

- Recall query rewriting:
  - **Mappings $\mathcal{M}$** *are* R2RML mappings
  - Run mappings **in reverse** to extract and transform data



- "**Lazy ETL**"
  - Deploy query rewriting (OBDA) system
  - Extend $\mathcal{O}$ and $\mathcal{M}$ as needed
  - Use $\mathcal{M}$ to ETL data into RDF store

Materialisation — Issues

1 Scalability

- Ptime complete
- Efficiently implementable in practice?

2 Updates

- Additions relatively easy (continue materialisation)
- But what about retraction?

3 Migrating data to RDF

- Materialisation assumes data in "special" (RDF triple) store
- How can legacy data be migrated?

4 Expressivity

- $\mathsf{QL} \not\subseteq \mathsf{RL}$; in particular, no RHS existentials (aka existential rules)

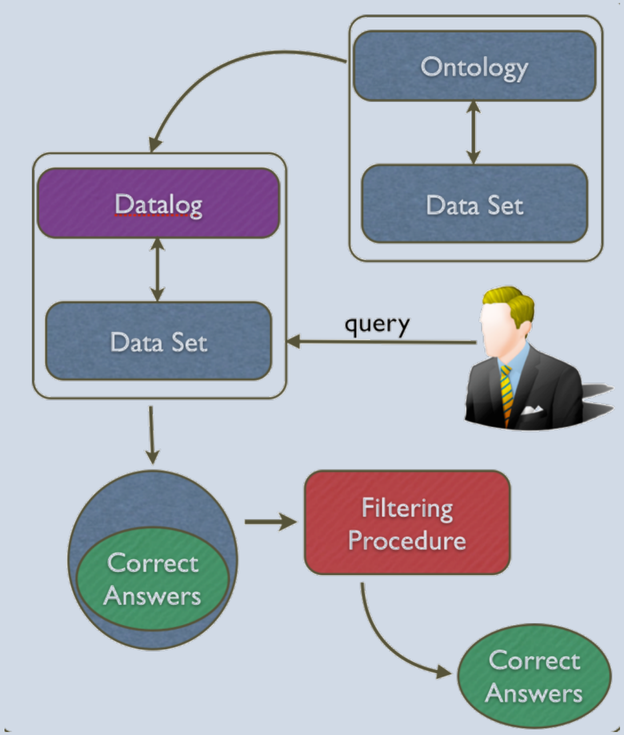
Materialisation: Expressivity

- RL is more powerful than QL, but $\mathsf{QL} \not\subseteq \mathsf{RL}$
  - In particular, no RHS existentials (aka existential rules)
  - Can't express, e.g., $\mathsf{OilPipeline} \sqsubseteq \mathsf{Pipeline} \sqcap \exists \mathsf{fromFacility}.\mathsf{OilFacility}$
- Recall **OWL 2 EL**
  - Based on $\mathcal{EL}^{++}$
  - Implementable via Datalog query answering plus "filtration"

OWL 2 EL via Datalog + Filtration

Given (RDF) Data Set, EL ontology $\mathcal{O}$ and query $\mathcal{Q}$:

- **Over-approximate** $\mathcal{O}$ into Datalog program D
- **Evaluate** $\mathcal{Q}$ over D + Data Set (e.g., via materialisation)
- Use (polynomial) **Filtering Procedure** to eliminate spurious answers



Discussion

- **QL-Rewriting** has many advantages
  - Data can be left untouched and in legacy storage
  - Exploits existing DB infrastructure and scalability
  - …
- But what if **more expressiveness/flexibility** is needed?
  - Query answering for EL and RL still tractable (polynomial)
  - Critically depend on Datalog scalability – RDFox to the rescue!
  - Easy migration path from QL-rewriting via “lazy ETL”

Future Work

- Piloting, evaluation and tuning
- Porting to other large-scale architectures
- Semantic (data) partitioning for distributed architectures
- (Incremental maintenance of) aggregations
- Improved query planning
- Stream reasoning
- Hybrid rewriting/materialisation (on demand) approach
- Expressiveness beyond RL/EL via PAGOdA techniques
- …

Acknowledgements

EPSRC Engineering and Physical Sciences Research Council

SIXTH FRAMEWORK PROGRAMME

SEVENTH FRAMEWORK PROGRAMME

DARPA

UNIVERSITY OF OXFORD
DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

EPSRC
Engineering and Physical Sciences Research Council

Optique

谢谢！

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE

Information Systems Group

EPSRC Engineering and Physical Sciences Research Council

Optique

谢谢！

欢迎提问！

UNIVERSITY OF OXFORD DEPARTMENT OF COMPUTER SCIENCE
Information Systems Group
EPSRC Engineering and Physical Sciences Research Council
Optique