

# Optique™



Scalable End-user Access to Big Data

**SIEMENS**



**fluidOps**



**UiO : University of Oslo**



**UNIVERSITÄT ZU LÜBECK**



**SAPIENZA**  
UNIVERSITÀ DI ROMA

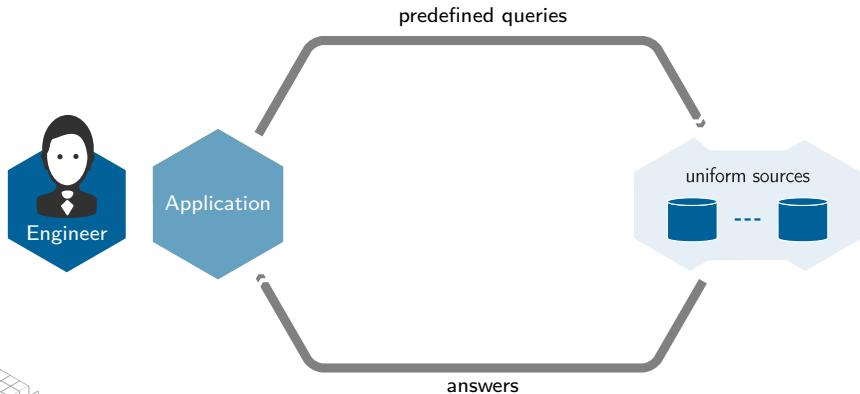


FREIE UNIVERSITÄT BOZEN  
LIBERA UNIVERSITÀ DI BOLZANO  
FREE UNIVERSITY OF BOZEN · BOLZANO



HELLENIC REPUBLIC  
**National and Kapodistrian  
University of Athens**

# The Problem of **Data Access**





I don't know how to formulate my information need in the application, it's too complex.



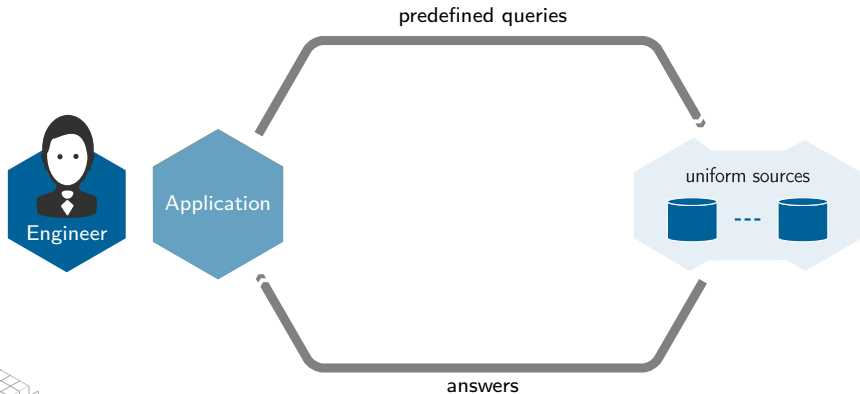
The application is not able to formulate my information need.



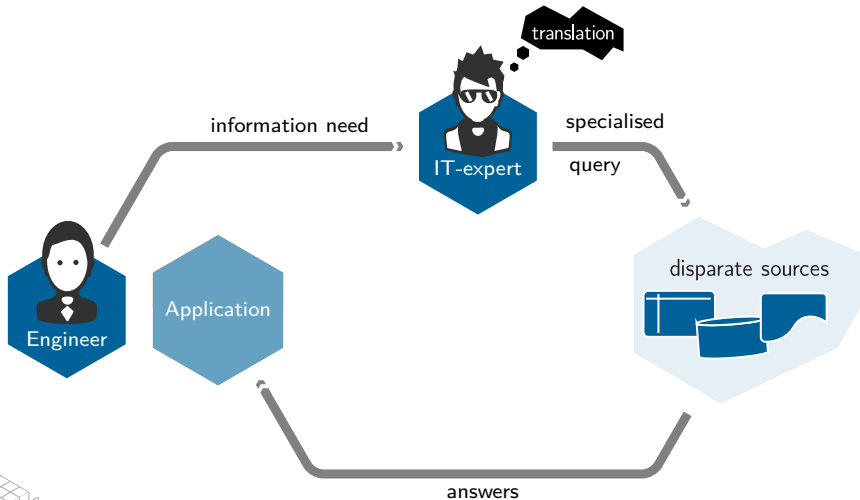
The system does not contain the data necessary to answer my information need.



# The Problem of **Data Access**

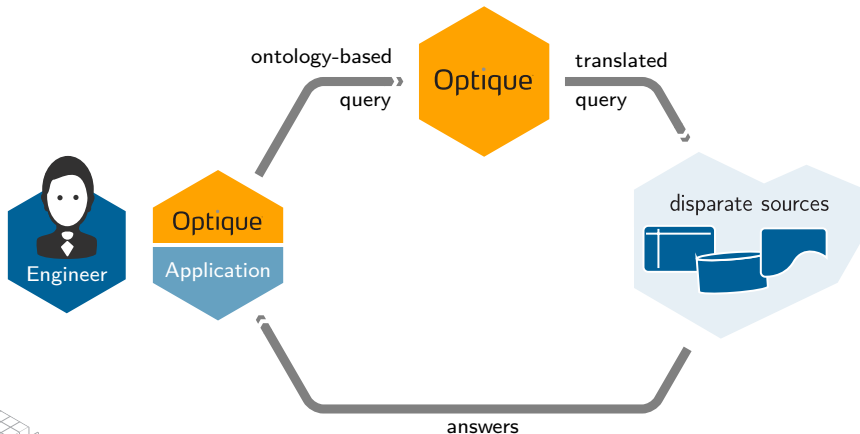


# The Problem of **Data Access**



# Data Access: The **Optique** Solution

Optique

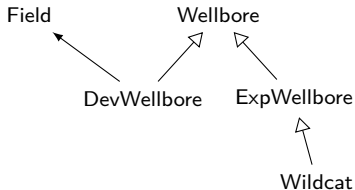


Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Query: *List all wellbores.*

Ontology:



Mappings:

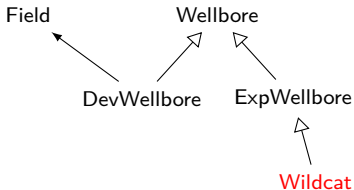
```
DevWellbore(name) ↦ SELECT name FROM wlb_dev  
ExpWellbore(name) ↦ SELECT name FROM wlb_exp  
Wildcat(name) ↦ SELECT name FROM wlb_exp  
                  WHERE purpose = 'WILDCAT'
```

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Query: *List all wellbores.*

Ontology:



Mappings:

```
DevWellbore(name) ↦ SELECT name FROM wlb_dev  
ExpWellbore(name) ↦ SELECT name FROM wlb_exp  
Wildcat(name) ↦ SELECT name FROM wlb_exp  
                  WHERE purpose = 'WILDCAT'
```



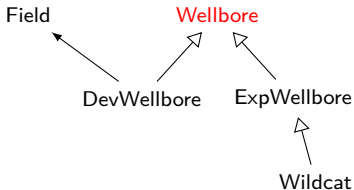
Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Query: *List all wellbores.*

q: Wellbore(x)

Ontology:



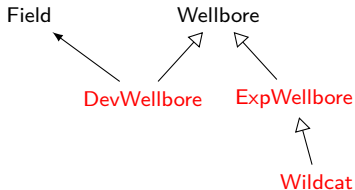
Mappings:

```
DevWellbore(name)  $\mapsto$  SELECT name FROM wlb_dev  
ExpWellbore(name)  $\mapsto$  SELECT name FROM wlb_exp  
Wildcat(name)  $\mapsto$  SELECT name FROM wlb_exp  
                  WHERE purpose = 'WILDCAT'
```

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Ontology:



Query: *List all wellbores.*

*q*: Wellbore(*x*)

*q<sub>0</sub>*: Wellbore(*x*)  $\cup$   
DevWellbore(*x*)  $\cup$   
ExpWellbore(*x*)  $\cup$   
Wildcat(*x*)

Mappings:

DevWellbore(*name*)  $\mapsto$  SELECT *name* FROM wlb\_dev

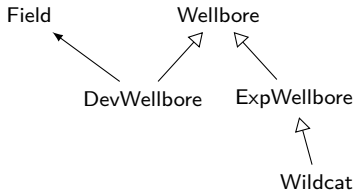
ExpWellbore(*name*)  $\mapsto$  SELECT *name* FROM wlb\_exp

Wildcat(*name*)  $\mapsto$  SELECT *name* FROM wlb\_exp  
WHERE purpose = 'WILDCAT'

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Ontology:



Mappings:

```
DevWellbore(name)  $\mapsto$  SELECT name FROM wlb_dev  
ExpWellbore(name)  $\mapsto$  SELECT name FROM wlb_exp  
Wildcat(name)  $\mapsto$  SELECT name FROM wlb_exp  
WHERE purpose = 'WILDCAT'
```

Query: *List all wellbores.*

*q*: Wellbore( $x$ )

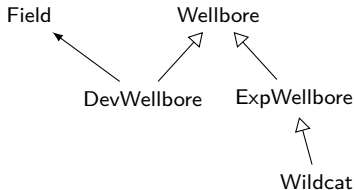
*q<sub>O</sub>*: Wellbore( $x$ )  $\cup$   
DevWellbore( $x$ )  $\cup$   
ExpWellbore( $x$ )  $\cup$   
Wildcat( $x$ )

*q<sub>SQL</sub>*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp  
UNION  
SELECT name FROM wlb\_exp  
WHERE purpose =  
'WILDCAT'

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Ontology:



Mappings:

```
DevWellbore(name)  $\mapsto$  SELECT name FROM wlb_dev  
ExpWellbore(name)  $\mapsto$  SELECT name FROM wlb_exp  
Wildcat(name)  $\mapsto$  SELECT name FROM wlb_exp  
WHERE purpose = 'WILDCAT'
```

Query: *List all wellbores.*

*q*: Wellbore(*x*)

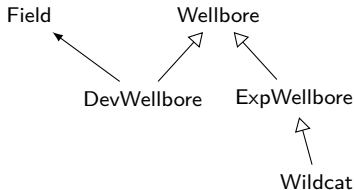
*q<sub>0</sub>*: Wellbore(*x*)  $\cup$   
DevWellbore(*x*)  $\cup$   
ExpWellbore(*x*)  $\cup$   
Wildcat(*x*)

*qSQL*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp  
UNION  
SELECT name FROM wlb\_exp  
WHERE purpose =  
'WILDCAT'

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Ontology:



Mappings:

```
DevWellbore(name)  $\mapsto$  SELECT name FROM wlb_dev  
ExpWellbore(name)  $\mapsto$  SELECT name FROM wlb_exp  
Wildcat(name)  $\mapsto$  SELECT name FROM wlb_exp  
WHERE purpose = 'WILDCAT'
```

Query: *List all wellbores.*

*q*: Wellbore(*x*)

*q<sub>O</sub>*: Wellbore(*x*)  $\cup$   
DevWellbore(*x*)  $\cup$   
ExpWellbore(*x*)  $\cup$   
Wildcat(*x*)

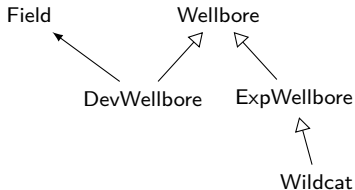
*q<sub>SQL</sub>*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp  
UNION  
SELECT name FROM wlb\_exp  
WHERE purpose =  
'WILDCAT'

*q'<sub>SQL</sub>*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp

Database:

```
wlb_dev(name, ...)  
wlb_exp(name, purpose, ...)
```

Ontology:



Mappings:

```
DevWellbore(name)  $\mapsto$  SELECT name FROM wlb_dev  
ExpWellbore(name)  $\mapsto$  SELECT name FROM wlb_exp  
Wildcat(name)  $\mapsto$  SELECT name FROM wlb_exp  
WHERE purpose = 'WILDCAT'
```

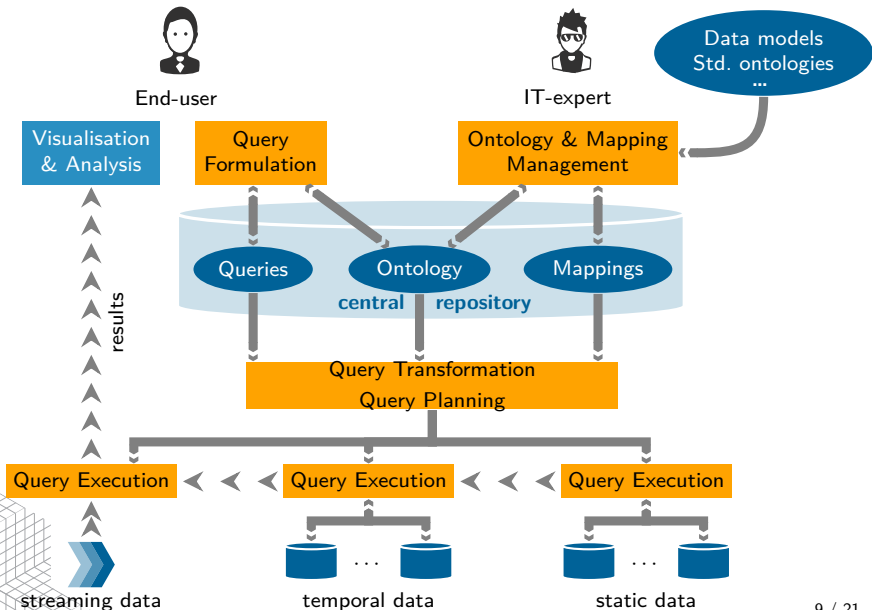
Query: *List all wellbores.*

*q*: Wellbore(*x*)

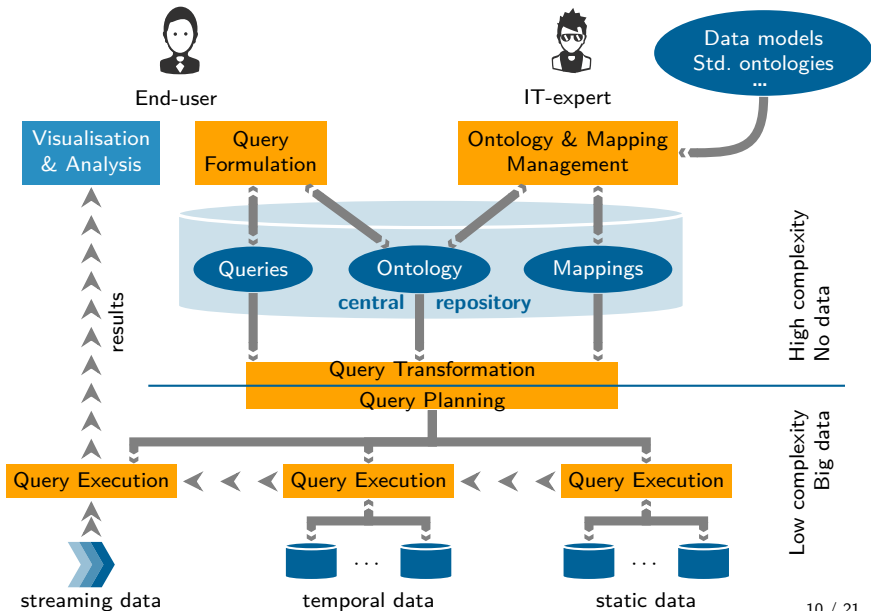
*q<sub>O</sub>*: Wellbore(*x*)  $\cup$   
DevWellbore(*x*)  $\cup$   
ExpWellbore(*x*)  $\cup$   
Wildcat(*x*)

*q<sub>SQL</sub>*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp  
UNION  
SELECT name FROM wlb\_exp  
WHERE purpose =  
'WILDCAT'

*q'<sub>SQL</sub>*: SELECT name FROM wlb\_dev  
UNION  
SELECT name FROM wlb\_exp



# Optique Architecture





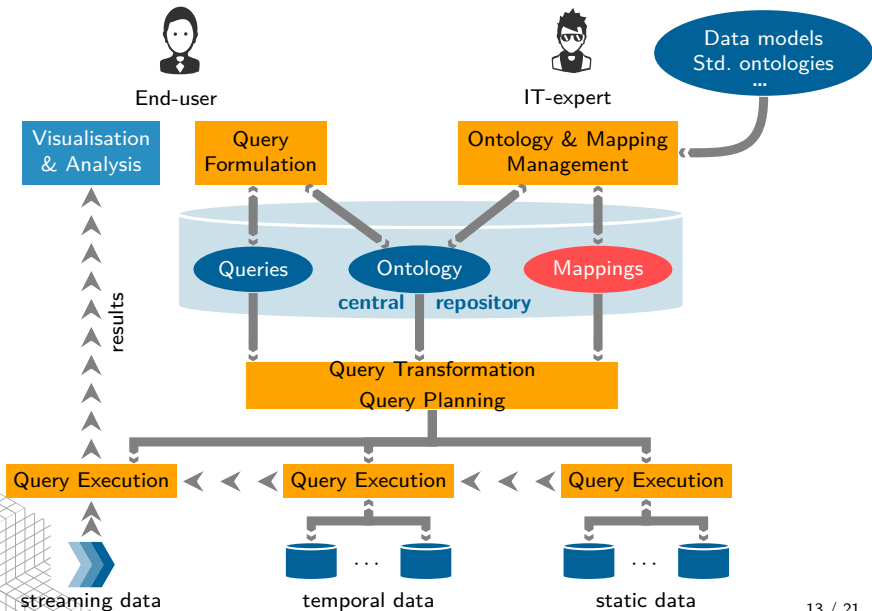
# OBDA is awesome!

OBDA on a single (large) datasource is awesome

- Generated SQL usually as good as manual
- The OBDA mappings are readable
- Visual Querying is possible

- Database keys must be treated perfectly
- Without keys, crucial optimizations are wrong
- Optional = Left join is hard
- Queries with slow user-defined SQL functions are hard to optimize

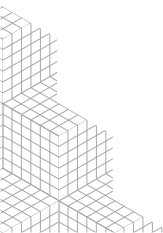




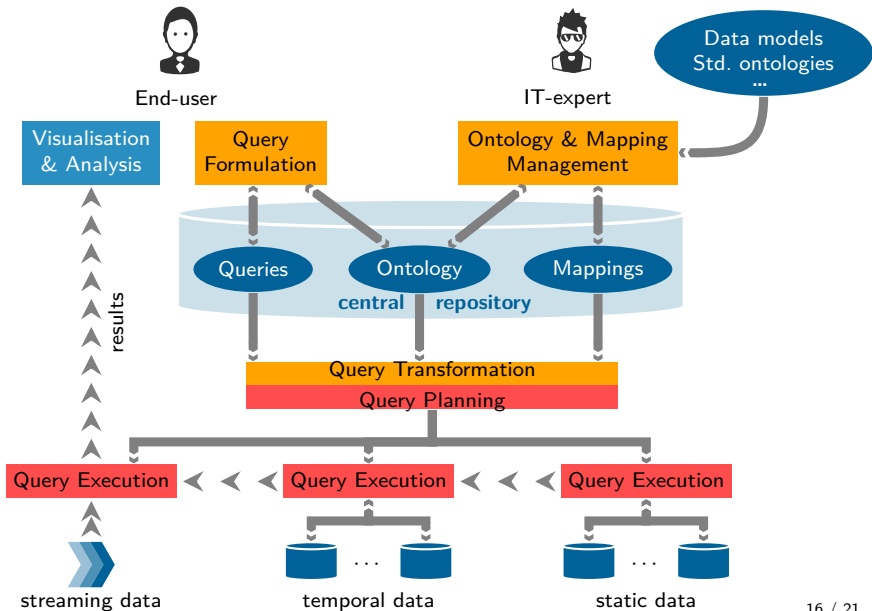
# OBDA Mapping SQL

- Readable SQL! (1-7 joins, average 2, compared to predefined queries with 10-90 joins)
- OBDA Mapping SQL: Minimal queries. Predefined SQL: Maximal queries

- The mappings have TBox and ABox URIs.
- TBox URI management is synchronized with ontology (aka easy)
- ABox URIs are only managed manually
- Errors in spelling or in use of columns for ABox URIs

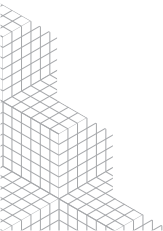


# Optique Architecture

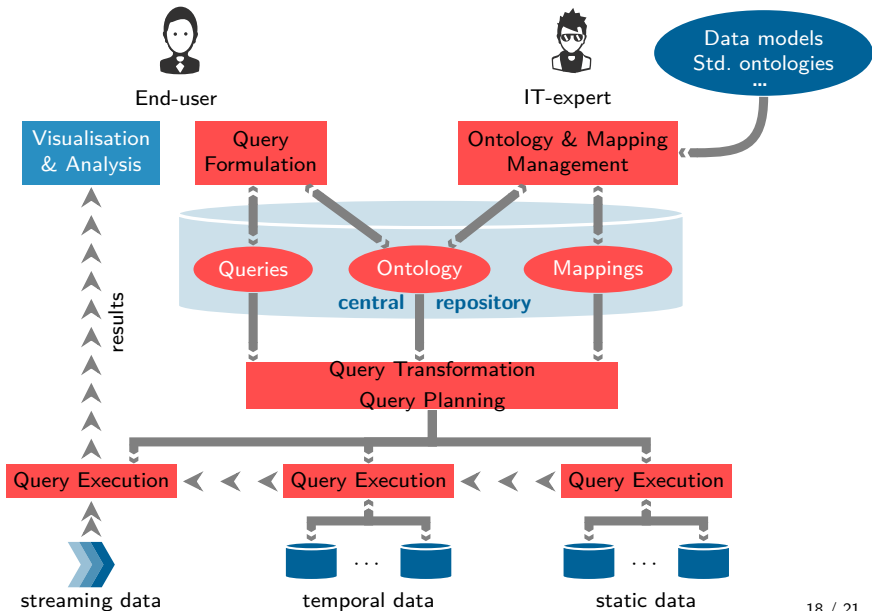


Goal: Querying multiple databases through a single SPARQL endpoint

- Executing SQL over several databases (Exareme)
- Connecting the same model to different databases (OBDA mappings and ontology axioms)
- Connecting equal objects in the different databases (Equality reasoning and management)



# Optique Architecture





# Managing Equality

- Managing equality information (not in Optique: heuristics? semi-manual?)
- Representing equality information in OBDA mappings
- Using *object* equality in OBDA rewriting (Ontop / owl:sameAs)

# Federated OBDA Mappings

- How to map concepts that cross database boundaries?
- Using the query editor: Writing it in SPARQL
- Using the mappings: Federated SQL queries
- But: No manual federated SQL allowed (for organisational reasons)
- Using the ontology: axioms and rules

# Authorization

- DB Authorization often connection-based
- Optique platform uses a single connection
- A connection per user?
- Authorization moved into platform?

Optique™

[www.optique-project.eu](http://www.optique-project.eu)