Optique

SEVENTH FRAMEWORK PROGRAMME

Scalable End-user Access to Big Data

SIEMENS

Statoil

fluid Operations

DNV·GL

UiO : University of Oslo

UNIVERSITY OF OXFORD

TUHH
Technische Universität Hamburg-Harburg

SAPIENZA
UNIVERSITÀ DI ROMA

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

HELLENIC REPUBLIC
National and Kapodistrian University of Athens

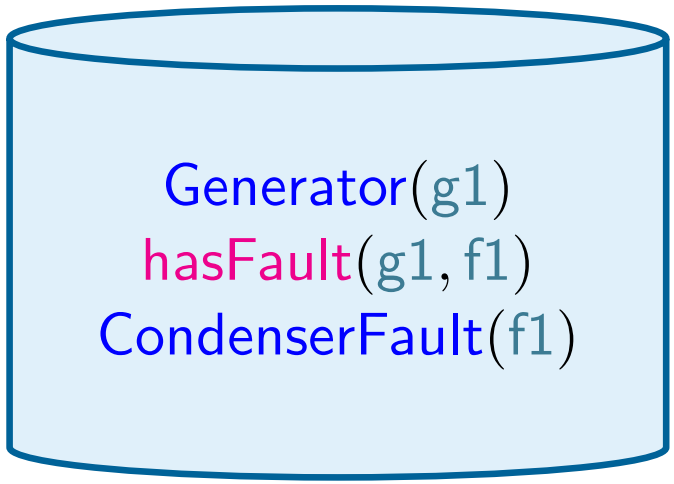
Ontology-based Data Access

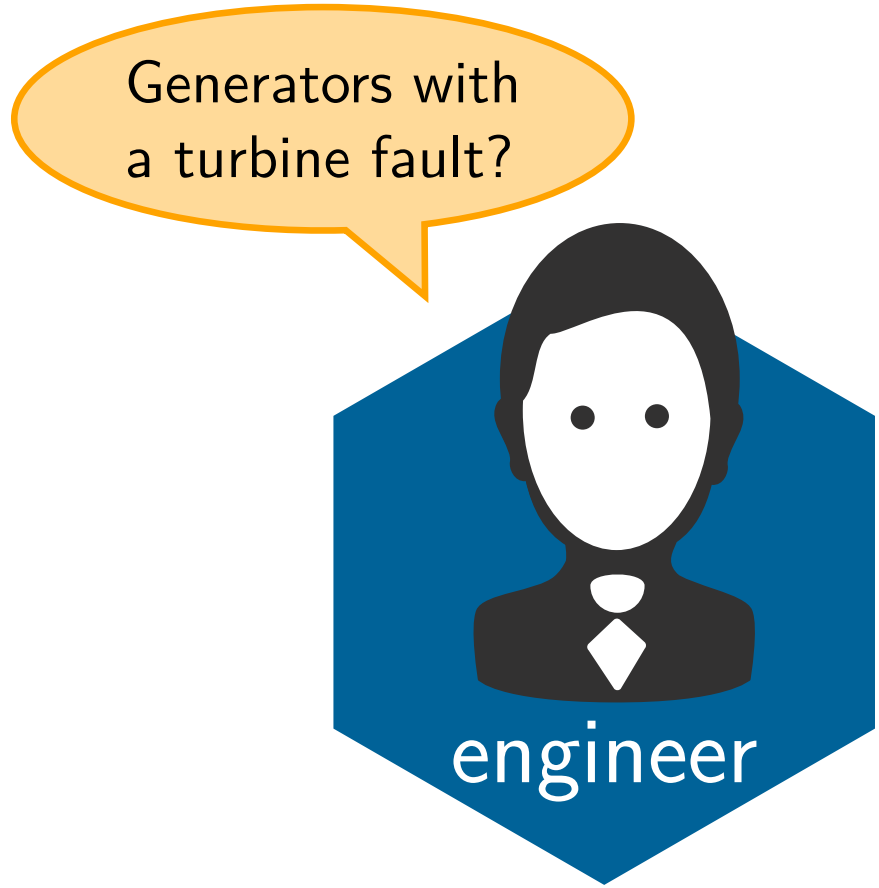
- Capture End-user vocabulary in an "Ontology"
  - $\approx$ Domain model
  - Classes and relations known to end-users
  - Some minimal domain knowledge
- Mappings that relate Ontology with data sources
  - 'Column "Type" is "T" in row x of table "Sensors" if sensor Nr. x is a Temperature Sensor'
- Automatically translate queries in End-user language to queries over data sources.

  In: 'List all temperature sensors.'

  Out: 'Print "Sensor Nr. x" for all rows x in "Sensors" table where "Type" column is "T."'

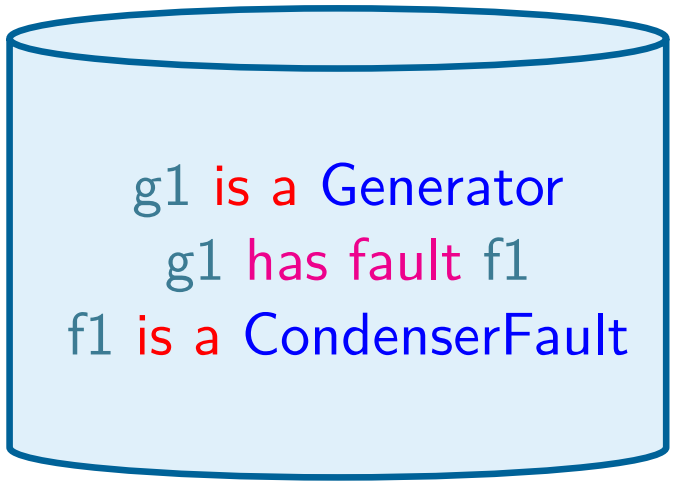
OBDA: **Example**

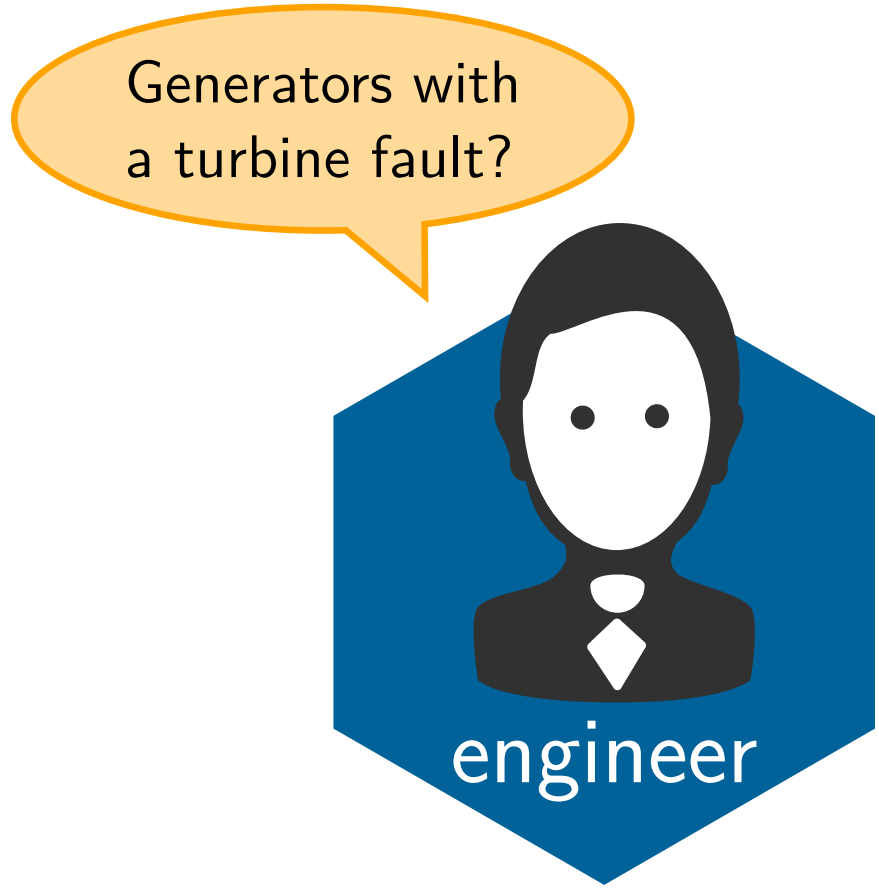




Based on slides by
Ian Horrocks

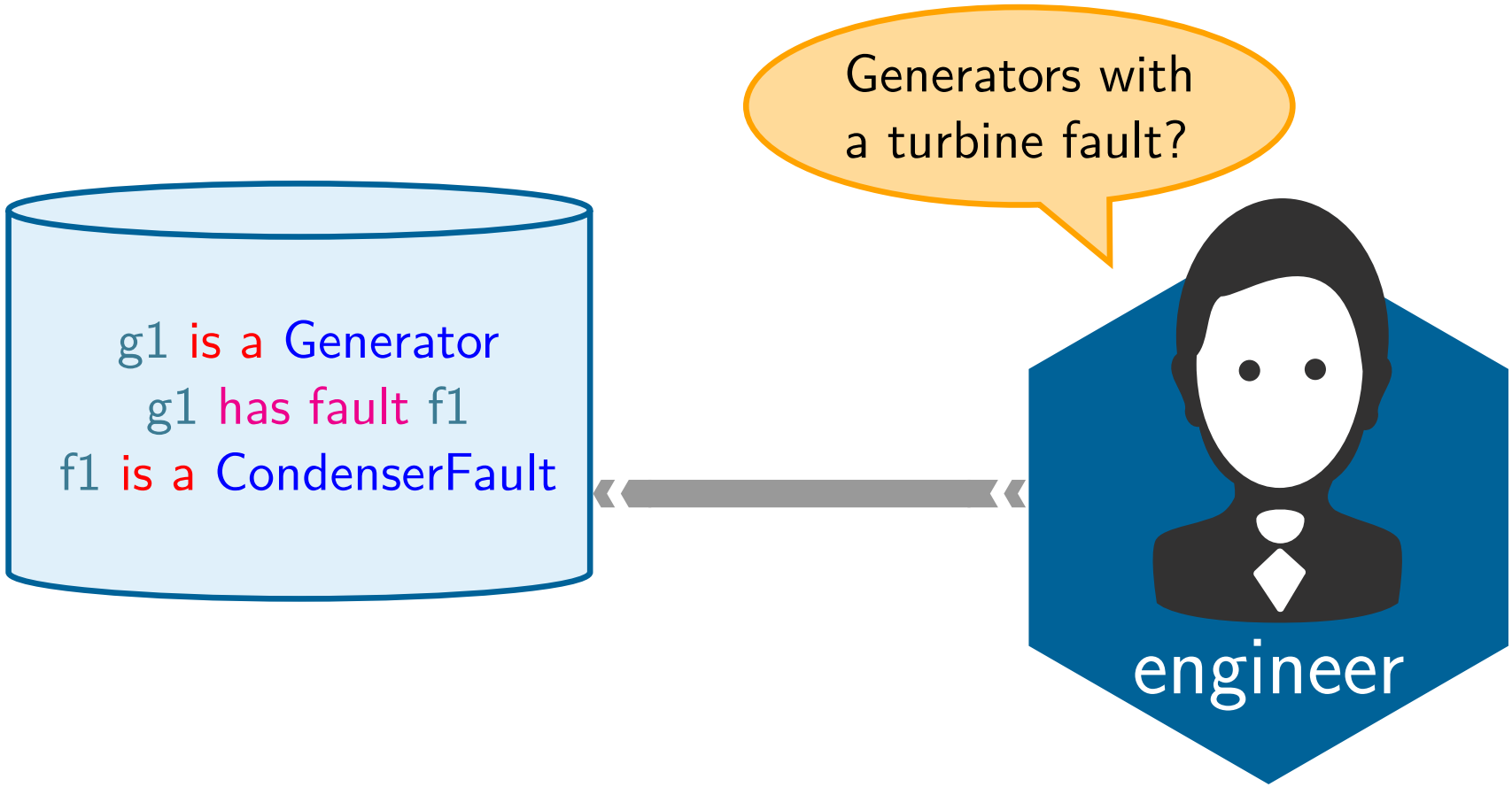
OBDA: Example





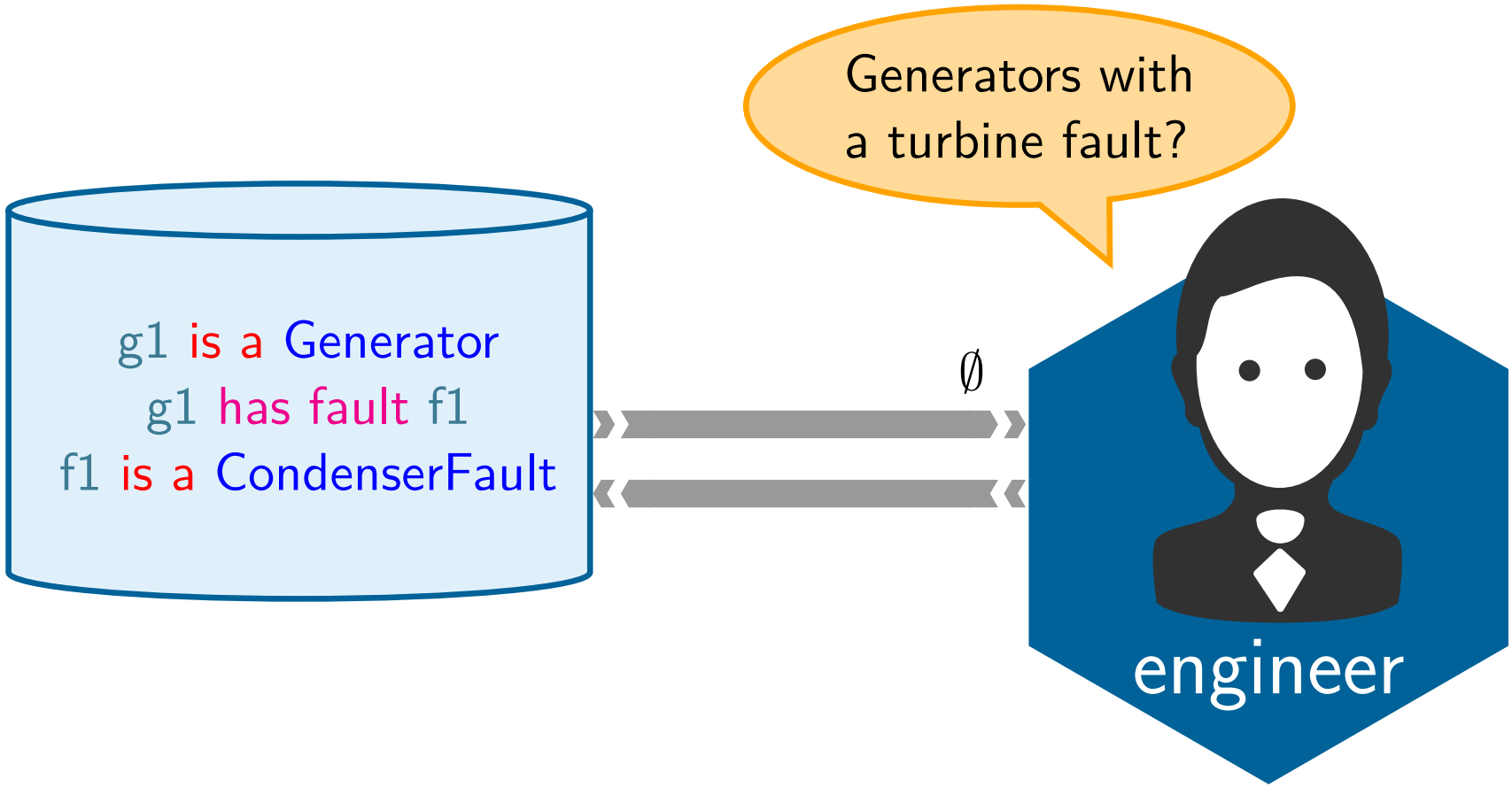
Based on slides by
Ian Horrocks

OBDA: **Example**



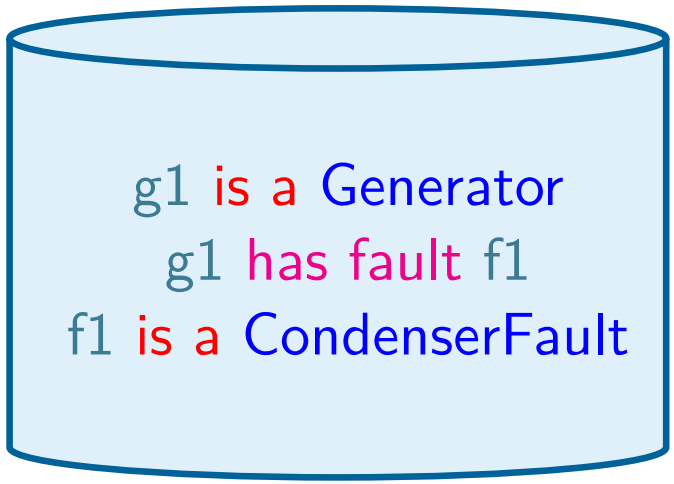
Based on slides by
Ian Horrocks

OBDA: **Example**

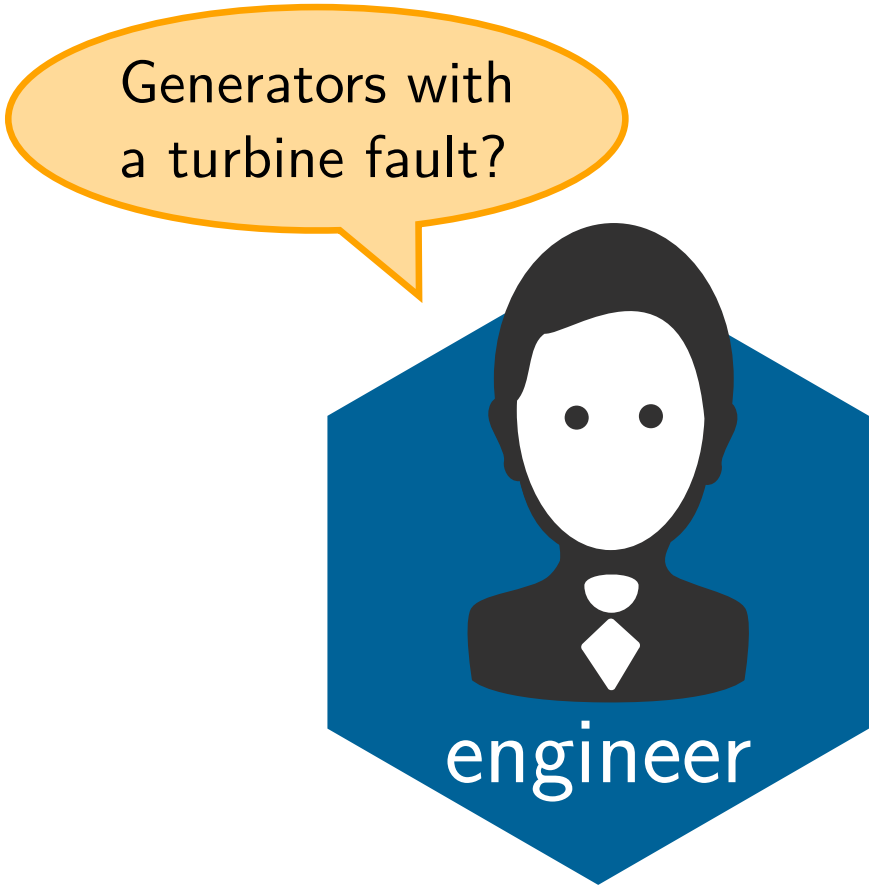


Based on slides by
Ian Horrocks

OBDA: **Example**

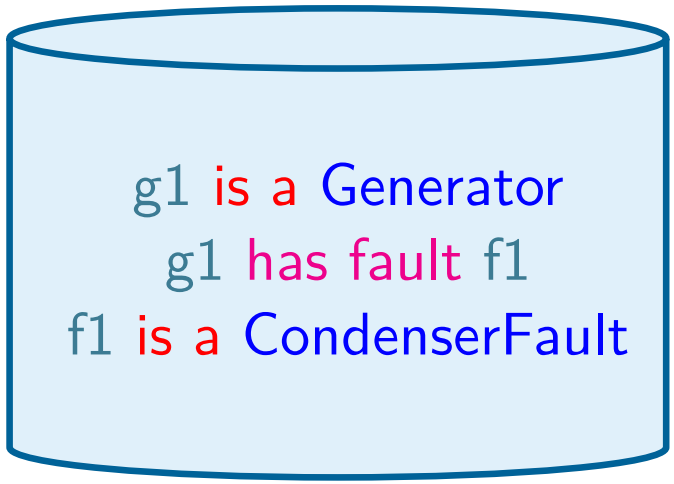


$$\begin{aligned}
\text{Condenser} &\sqsubseteq \text{CoolingDevice} \sqcap \exists \text{isPartOf}.\text{Turbine} \\
\text{CondenserFault} &\equiv \text{Fault} \sqcap \exists \text{affects}.\text{Condenser} \\
\text{TurbineFault} &\equiv \text{Fault} \sqcap \exists \text{affects}.(\exists \text{isPartOf}.\text{Turbine})
\end{aligned}$$



Based on slides by
Ian Horrocks

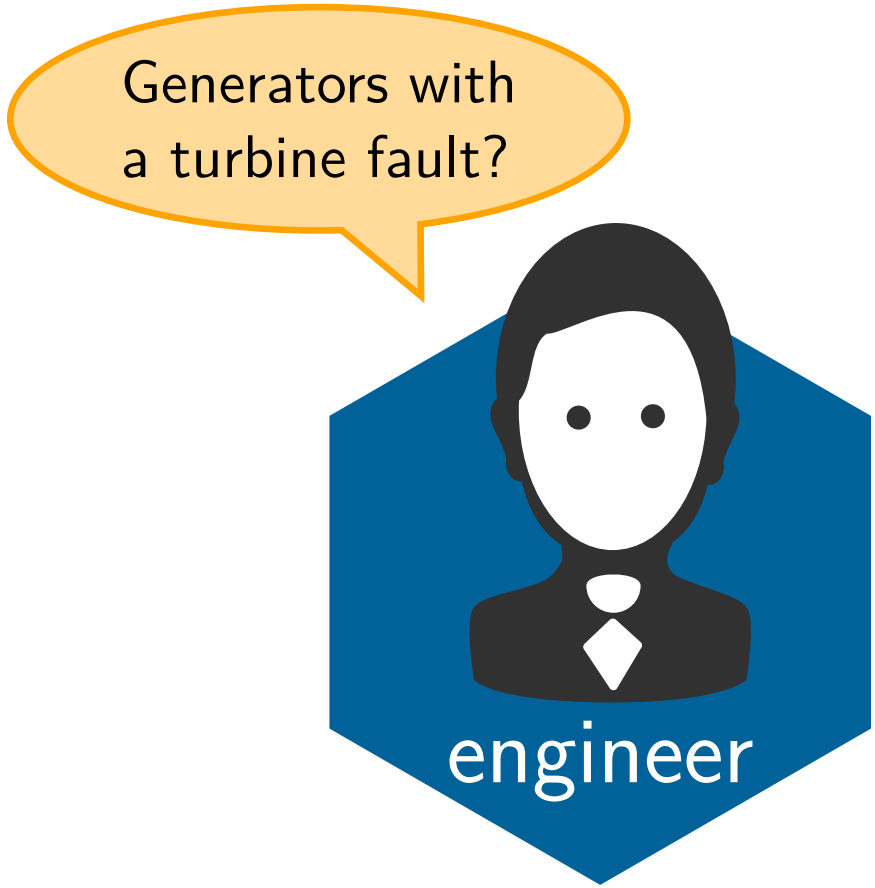
OBDA: **Example**



Condenser is a CoolingDevice that is part of a Turbine

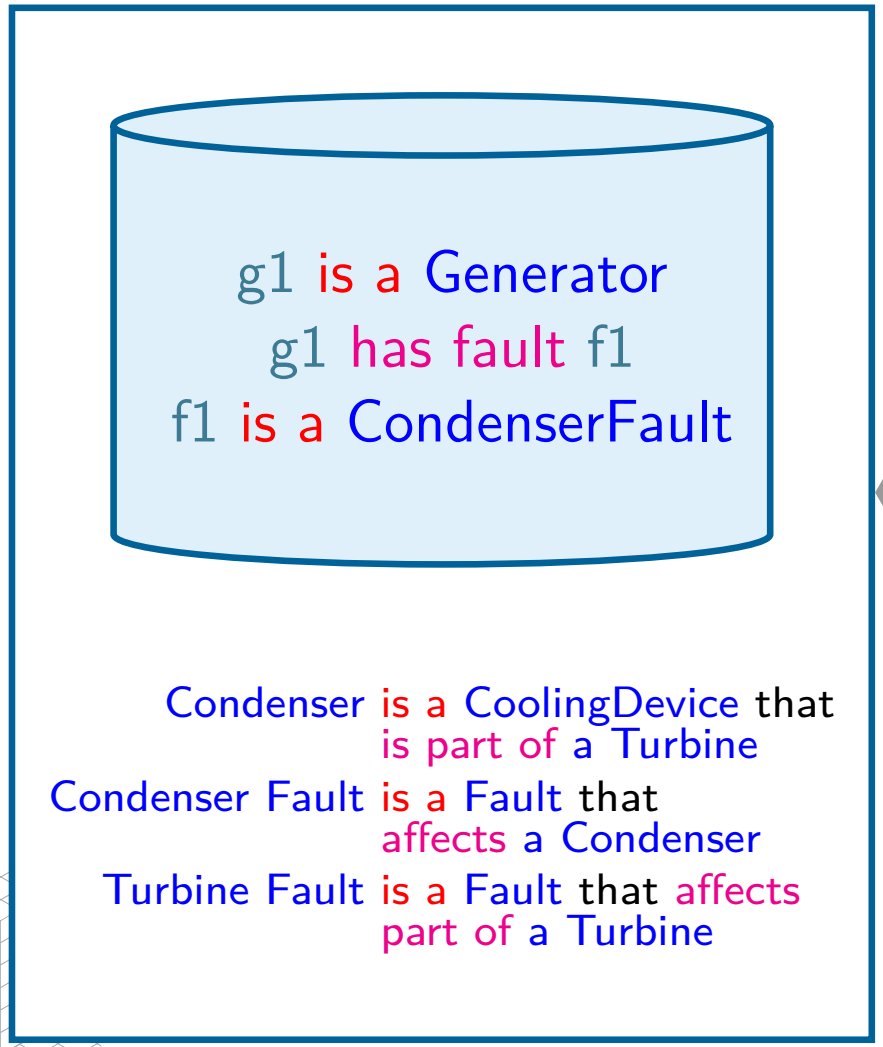
Condenser Fault is a Fault that affects a Condenser

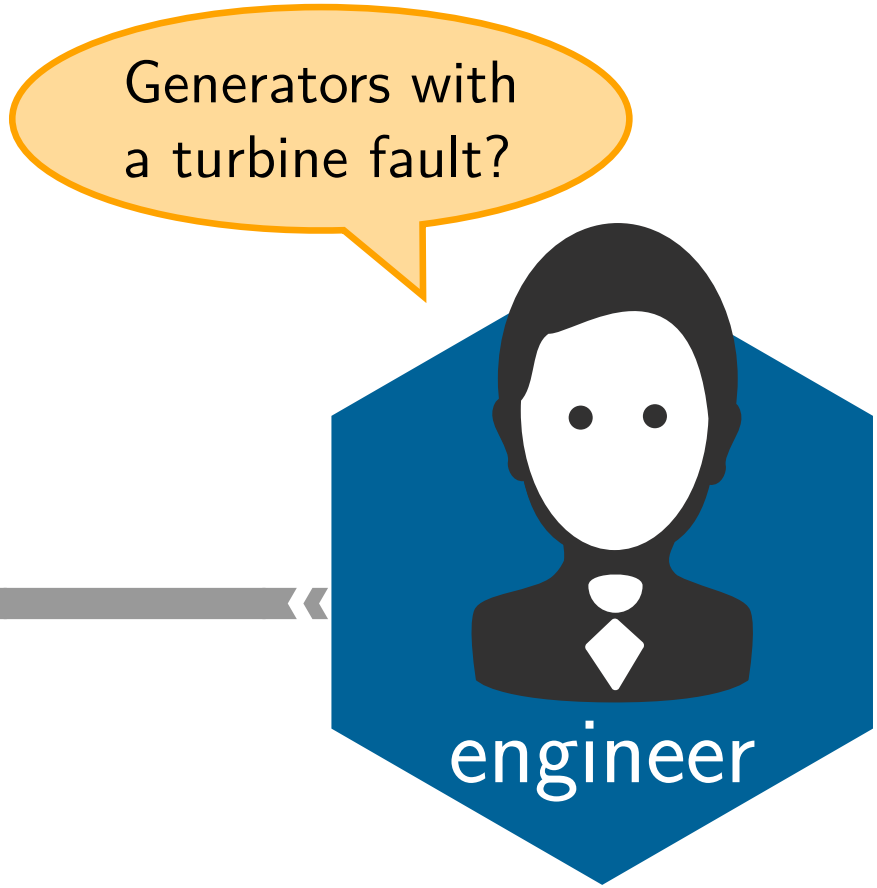
Turbine Fault is a Fault that affects part of a Turbine



Based on slides by Ian Horrocks

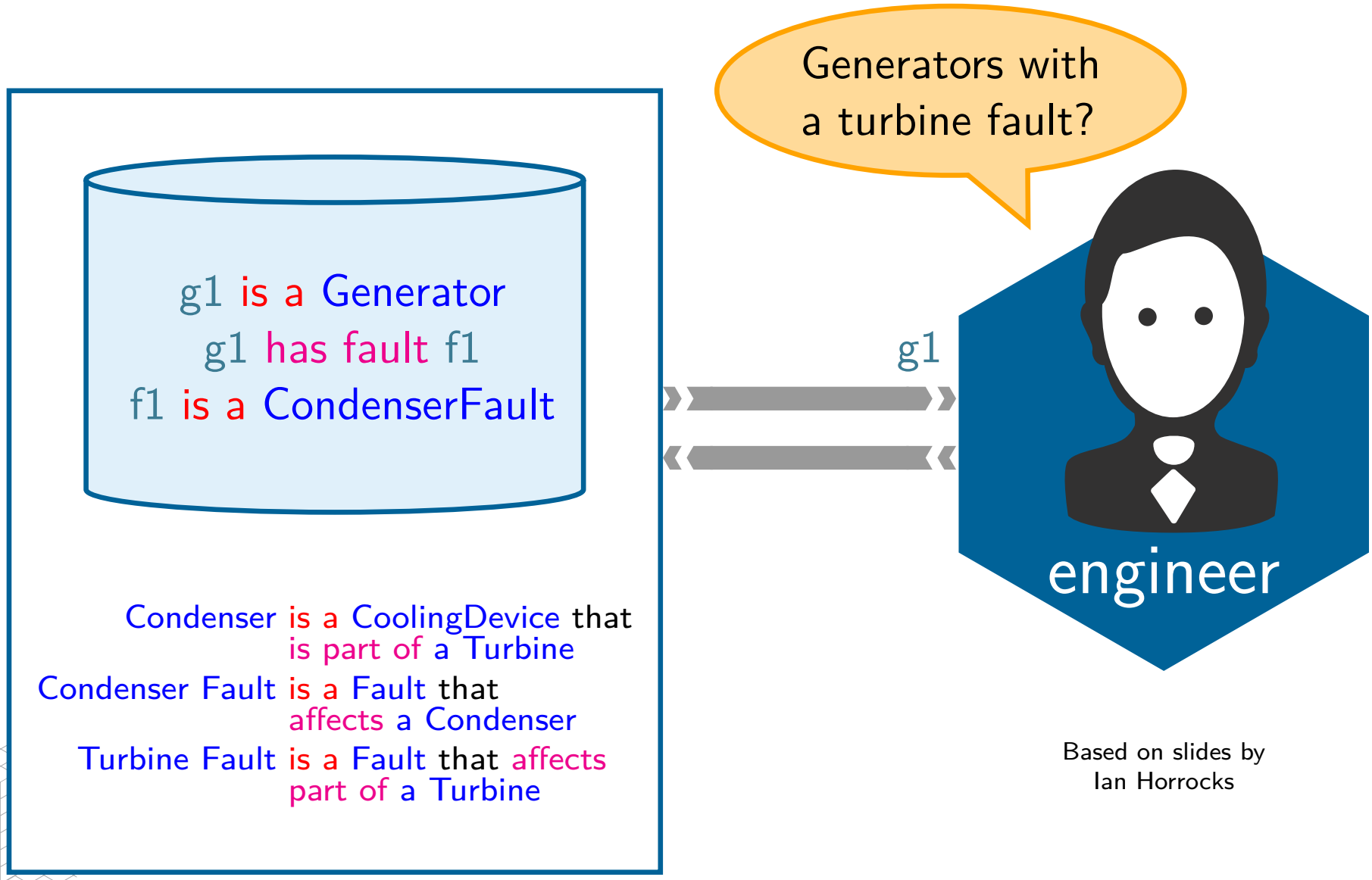
OBDA: **Example**





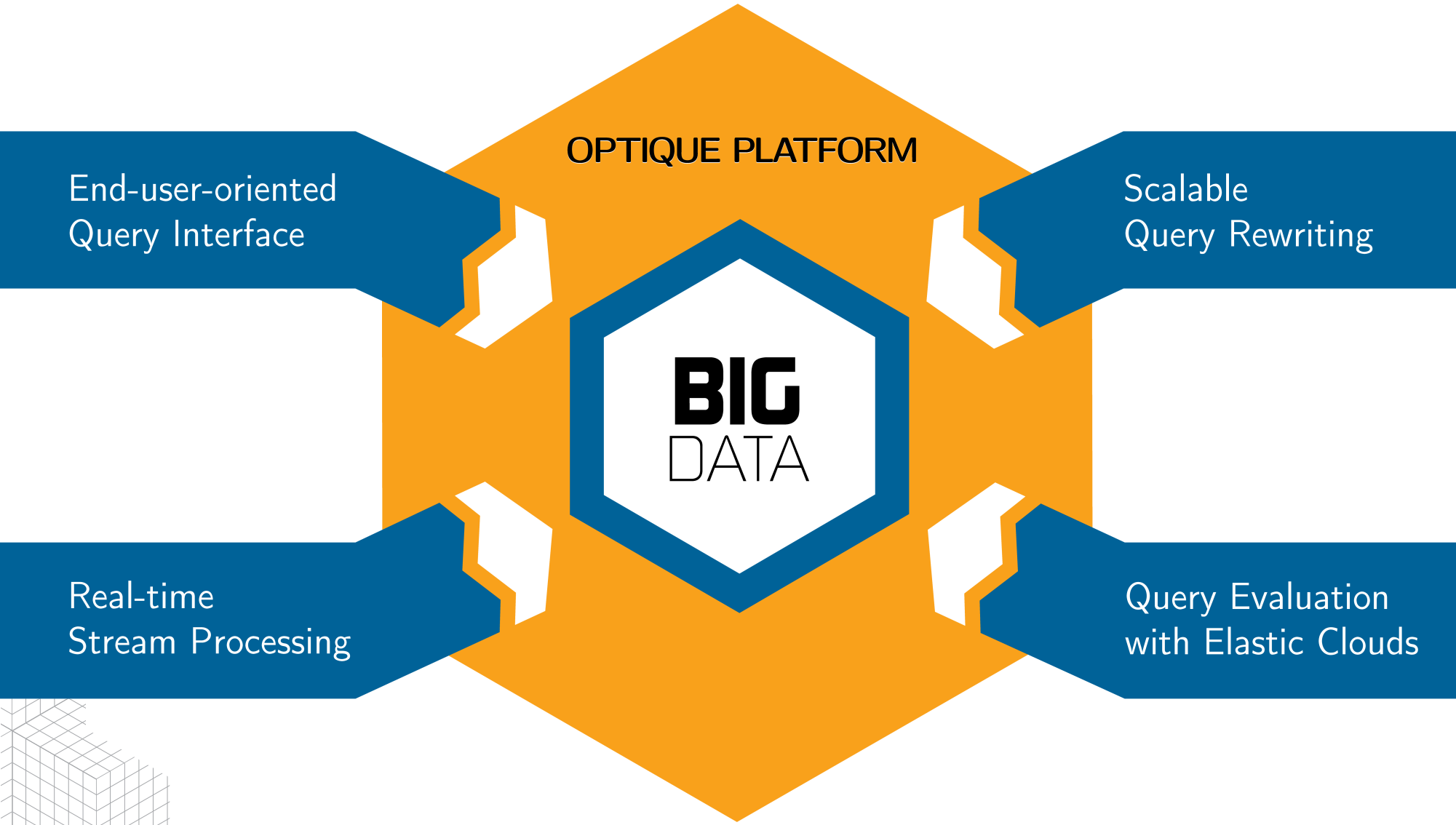
Based on slides by Ian Horrocks

OBDA: **Example**

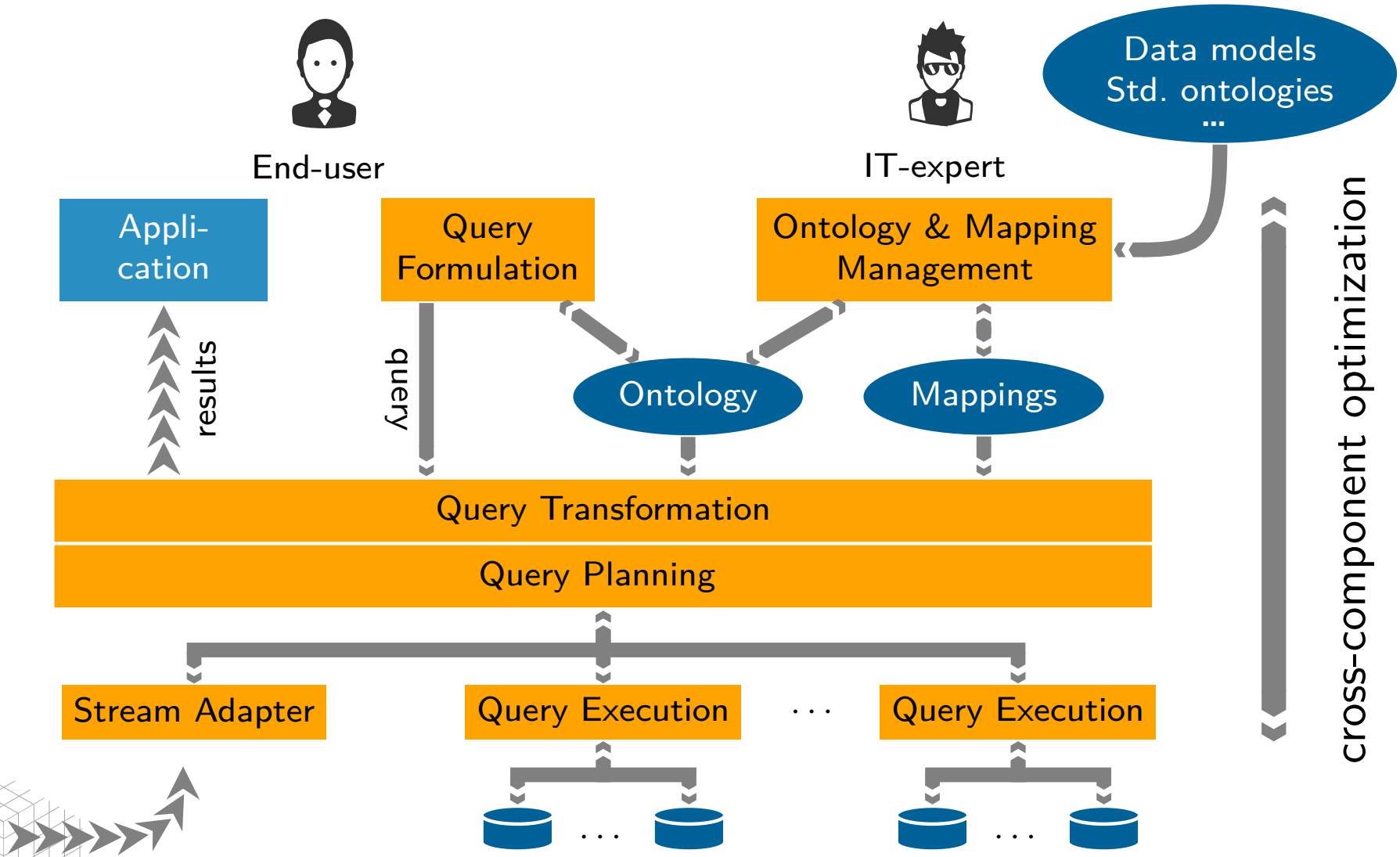


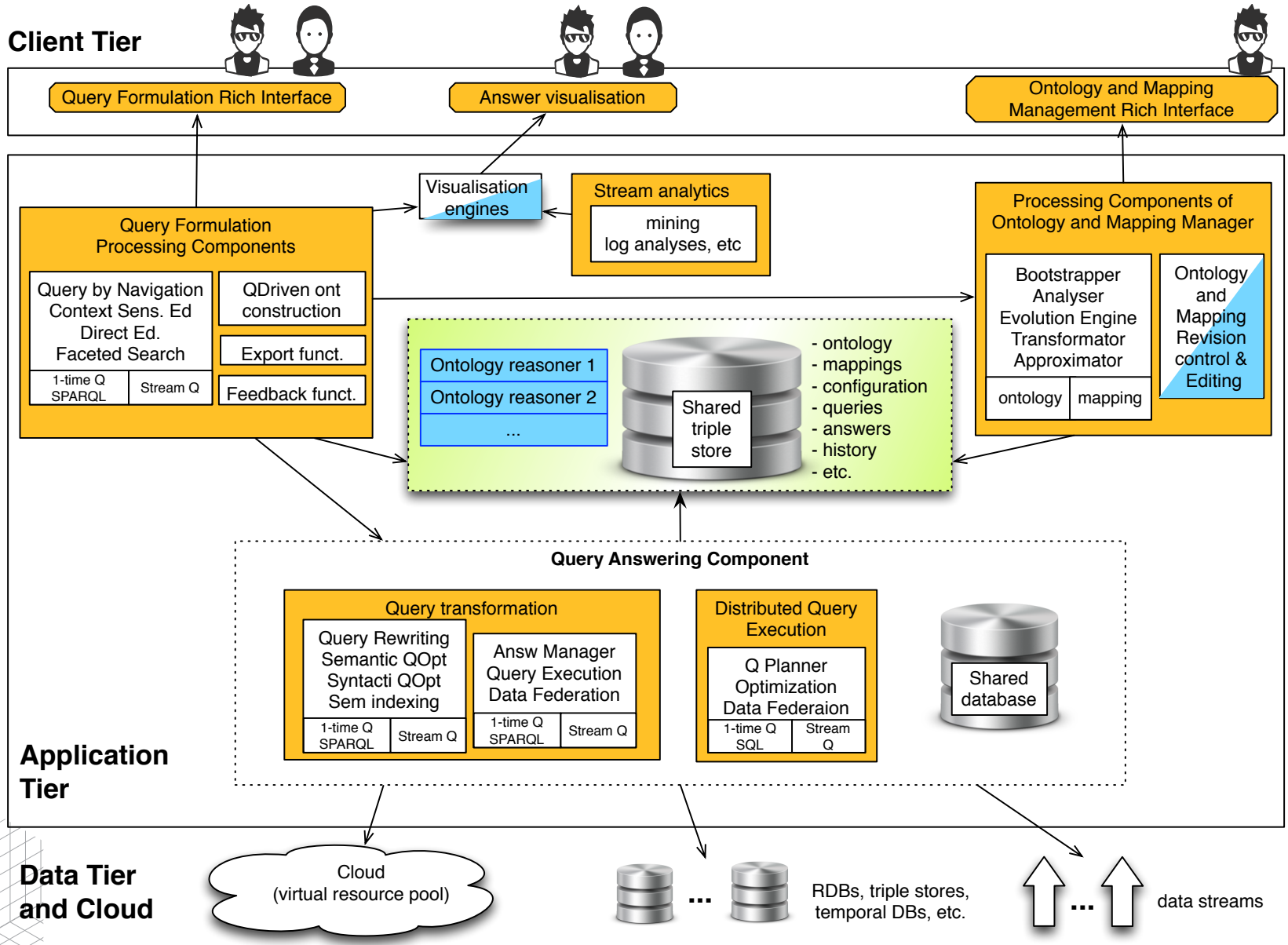
Based on slides by
Ian Horrocks

Unique Combination of Techniques



Optique Architecture



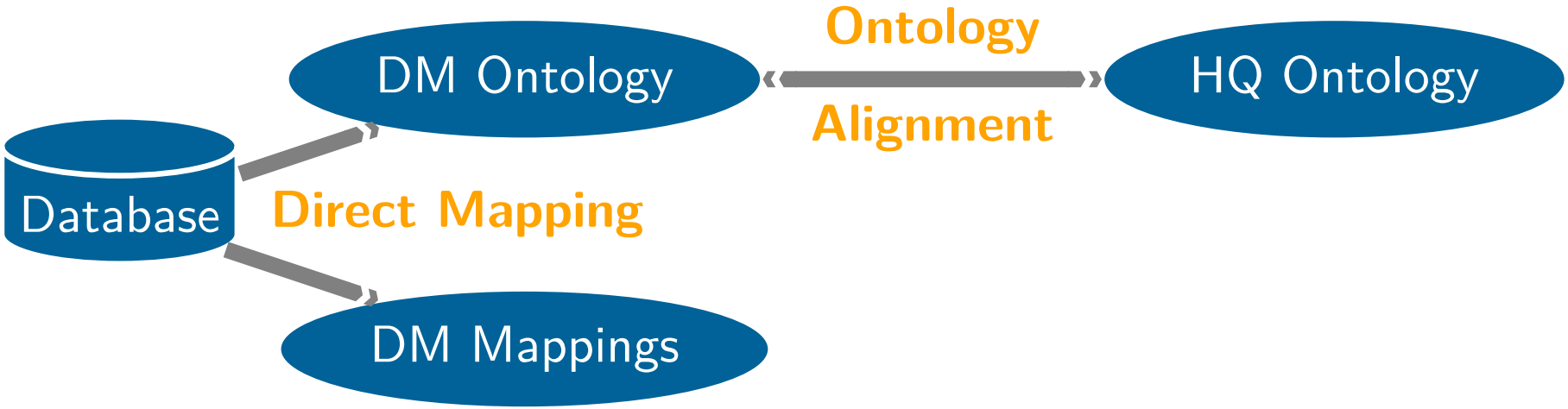

Client Tier
Query Formulation Rich Interface
Answer visualisation
Ontology and Mapping Management Rich Interface
Visualisation engines
Stream analytics
mining
log analyses, etc
Query Formulation Processing Components
Query by Navigation
Context Sens. Ed
Direct Ed.
Faceted Search
1-time Q SPARQL
Stream Q
QDriven ont construction
Export funct.
Feedback funct.
Processing Components of Ontology and Mapping Manager
Bootstrapper
Analyser
Evolution Engine
Transformator
Approximator
ontology
mapping
Ontology and Mapping Revision control & Editing
Ontology reasoner 1
Ontology reasoner 2
...
Shared triple store
- ontology
- mappings
- configuration
- queries
- answers
- history
- etc.
Query Answering Component
Query transformation
Query Rewriting
Semantic QOpt
Syntacti QOpt
Sem indexing
1-time Q SPARQL
Stream Q
Answ Manager
Query Execution
Data Federation
1-time Q SPARQL
Stream Q
Distributed Query Execution
Q Planner
Optimization
Data Federaion
1-time Q SQL
Stream Q
Shared database
Application Tier
Data Tier and Cloud
Cloud
(virtual resource pool)
...
RDBs, triple stores, temporal DBs, etc.
...
data streams

The **Query Formulation** Interface

- Let users formulate ad-hoc queries
  - filtering on attributes
  - connecting objects
  - selecting what information to extract
  - choosing types (Facility $\rightarrow$ FixedFacility | MovableFacility)
- Until end of year:
  - specify time ranges
  - choose entities (licenses, fields, etc.) from map
- Later:
  - aggregation: sums, averages, etc.
  - negation ("all turbines *without* a fault")
- Intentionally restricted expressivity
  - As powerful as SQL $\rightarrow$ as hard to learn
- Demo
  - Data from NPD FactPages (`http://factpages.npd.no/`)

Ontology & Mapping Management

- OBDA relies on Ontology and Mappings
- Tool support to create and maintain O&M
- Results so far: Bootstrapping components



- Coming up: tool support for O&M QC and evolution
  - when ontology changes
  - when data sources change

Time & Streams

- Query processing extended for stream queries (STARQL)
  - combined queries on real-time and historical data
  - rewrite queries over temporal data
  - execution with streaming answers in ADP (→ slide 11)

- Coming up: integration with platform architecture
  - register/unregister queries
  - stream answers
  - (also useful for one-shot queries)

Query Transformation

- Based on open source -ontop- system
  - Query rewriting for OWL 2 QL ontologies
  - Covers almost all of standard SPARQL query language
- Now testing on real queries from Statoil on EPDS
  - Efficiency problems with some rewritten queries
  - Targeted optimisation based on use-case requirements

Distributed Query Execution

- Query Execution ("backend")
  - Based on ADP – Athena Distributed Processing
  - Cutting edge parallelised database engine
  - Optimisation w.r.t. many dimensions
  - "Hadoop for Databases"
- For Optique:
  - stream processing
  - federation (one query, many sources)
  - parallelisation (elastic clouds)
- Cross-component optimisation of query processing

www.optique-project.eu