Ontology-Based Integration of Streaming & Static RDBs

Optique

E. Kharlamov, S. Brandt, M. Giese, E. Jimenez-Ruiz, Y. Kotidis, S. Lamparter, T. Mailis, C. Neuenstadt, O. Oezcep, C. Pinkel, A. Soylu, C. Svingos, D. Zheleznyakov, I. Horrocks. Y. Ioannidis, R. Moeller, A. Waaler

fluidOps SIEMENS
HELLENIC REPUBLIC National and Kapodistrian University of Athens

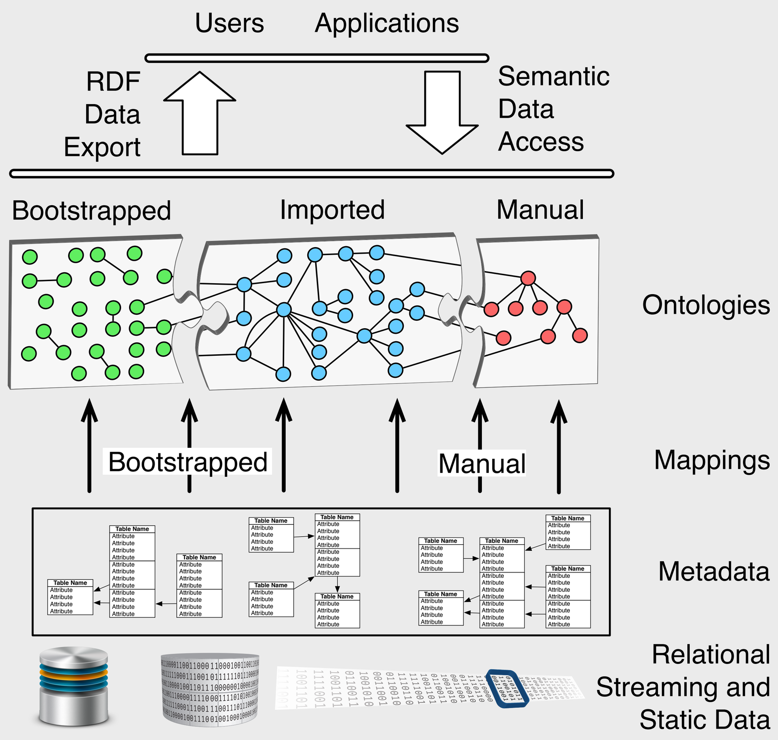
Semantic Access to Databases

Large enterprise databases

- Many complex different schemata
- Siemens
  - about 100s turbines produce data
  - life, archived streams, static RDBs
  - data access is hard: up to 80% of analytics time

Ontology Based Data Access

- Ontology: conceptual domain model
- Mappings: ontological terms to DBs



Research Challenges

Deployment support

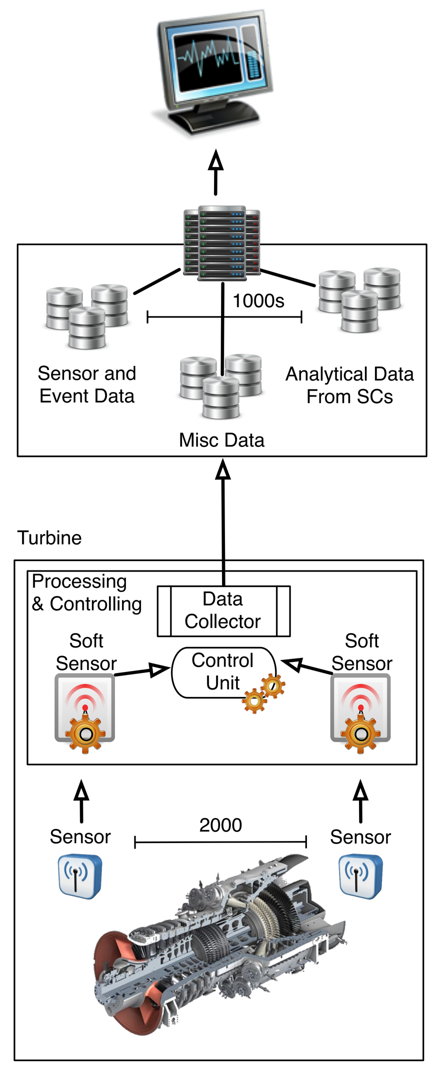
- semi-automatic for ontologies and mappings

Query language

- over ontologies, streaming and static data
- efficient query enrichment and transformation

Backend

- to optimise large numbers of queries
- efficiently execution over distributed streaming and static data



Diagnostic Queries with STARQL

```
PREFIX ex : <http://www.siemens.com/onto/gasturbine/>

CREATE PULSE examplePulse WITH START = NOW, FREQUENCY = 1min

CREATE STREAM StreamOfSensorsInCriticalMode AS
CONSTRUCT GRAPH NOW { ?sensor a :InCriticalMode }

FROM STATIC ONTOLOGY ex:sensorOntology, DATA ex:sensorStaticData
WHERE { ?sensor a ex:Reliable }

FROM STREAM    sensorMeasurements             [NOW - 1min, NOW]-> 1sec
               referenceSensorMeasurements 1year <-[NOW - 1min, NOW]-> 1sec,
USING PULSE    examplePulse
SEQUENCE BY    StandardSequencing AS MergedSequenceOfMeasurementes
HAVING EXISTS i IN MergedSequenceOfMeasurementes
      (GRAPH i { ?sensor ex:hasValue ?y. ex:refSensor ex:hasValue ?z })
      HAVING PearsonCorrelation(?y, ?z) > 0.75
```

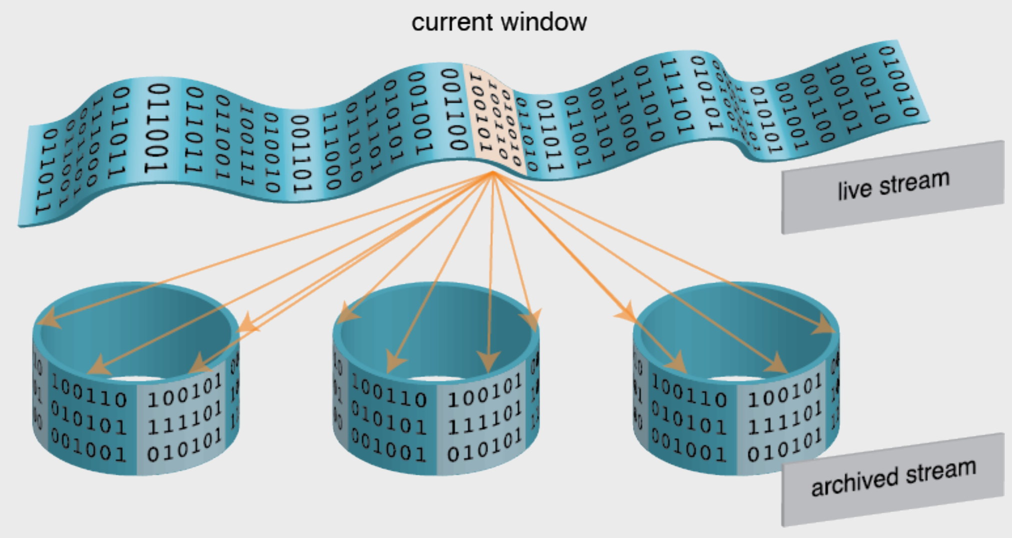
Main Features of STARQL

- Query language over ontologies
- Syntax: extension of SPARQL
  - basic graph patterns
  - typical mathematical, statistical, and event pattern features needed in real-time diagnostic scenarios
- Semantics
  - combination of open and closed world reasoning
  - extends snapshot semantics for window operators with sequencing semantics that can handle integrity constraints such as functionality assertions
- Efficient query enrichment and transformation
  - enrichment: PTime in the size of OWL 2 QL ontology
  - unfolding: in EXASTREAM hybrid queries

Stream-Static Query Processing with EXASTREAM

Main Features

- Highly optimised query processing system
- Supported queries
  - Extension of SQL
  - Hybrid stream-static
- High-throughput

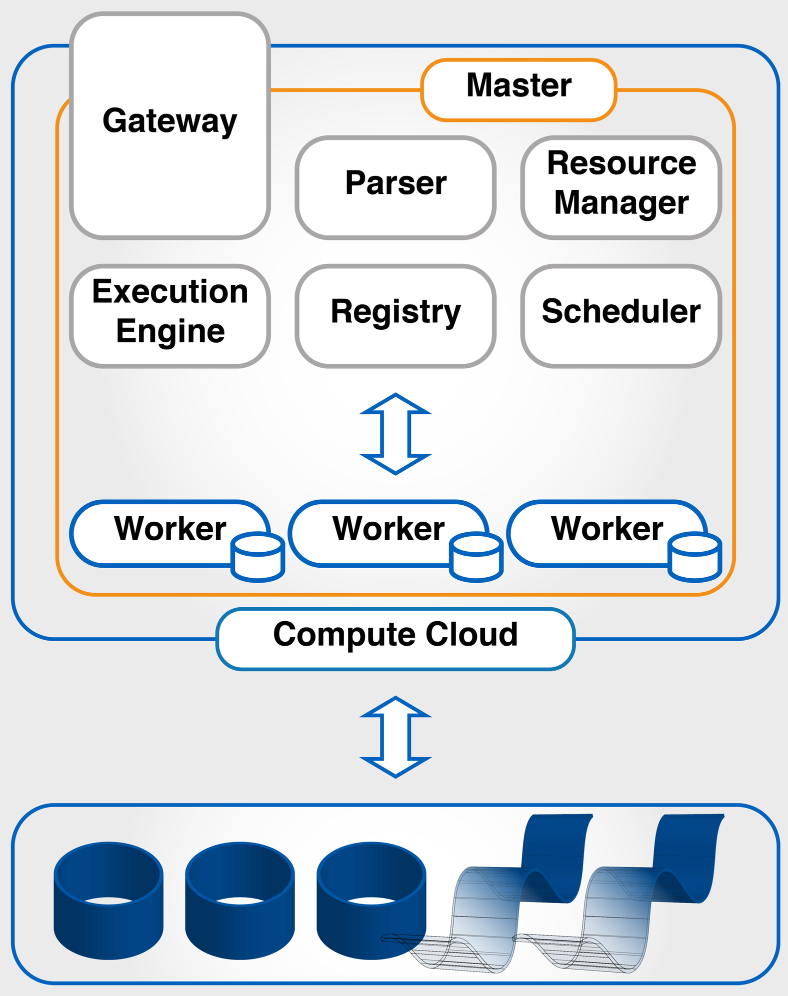


User Defined Functions

- For complex stream processing
- Arbitrary user code

Architecture

- Parallelism by distributing Q. processing across multiple nodes
- Query preprocessing
  - registered at *Gateway Server*
  - passed through *Parser*
  - fed into *Scheduler*
- Query execution
  - *Scheduler* finds *Worker Nodes* based on their load
  - *Scheduler* places stream & relations on selected *Workers*
  - *Worker Nodes* execute queries



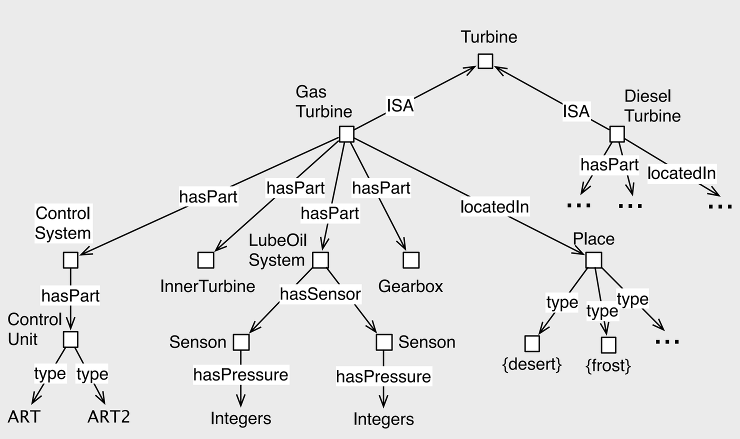
Demo Scenario

Demo Description

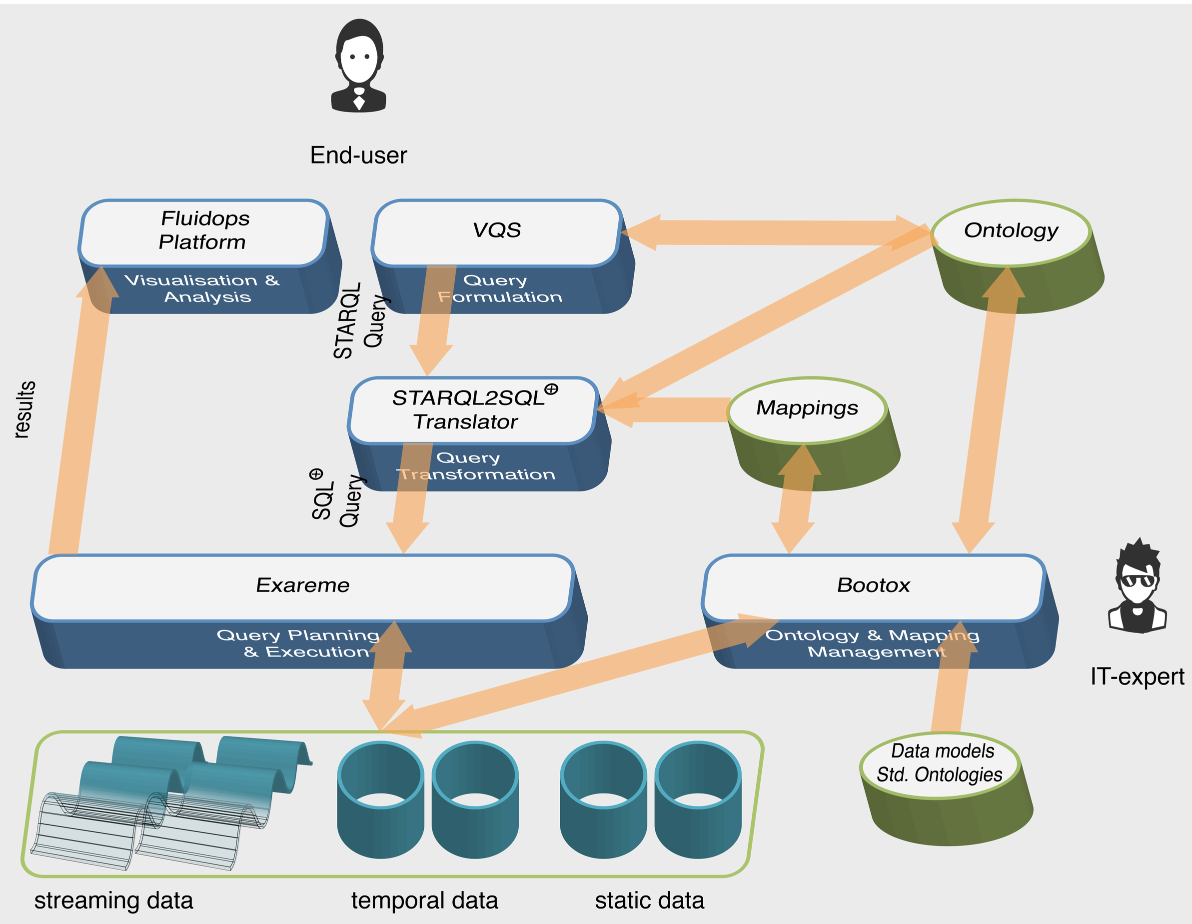
- Siemens diagnostics tasks
  - e.g., calculate the Pearson correlation coefficient between turbine data streams
- Siemens data
  - 950 turbines, 2002 – 11 years
  - anonumised
- Data distribution
  - from 1 to 128 nodes
  - each node: 2 proc., 4GB RAM

Demo Scenarios

- Diagnostics with our deployment
- Performance showcase of our deployment
- Diagnostics with user's deployment



Optique Platform



Main features

- End-to-end OBDA system
  - fully integrated
- For IT specialists
  - the whole OBDA life cycle
  - flexible configuration
- For end-users
  - intuitive query formulation
  - monitoring dash-boards
  - integration with GIS systems

OBDA query answering

Q1: Onto Query → Enrichment → Q2: Enriched Onto Query → Transformation → Q3: DB Query → DB Answers → Onto Answers