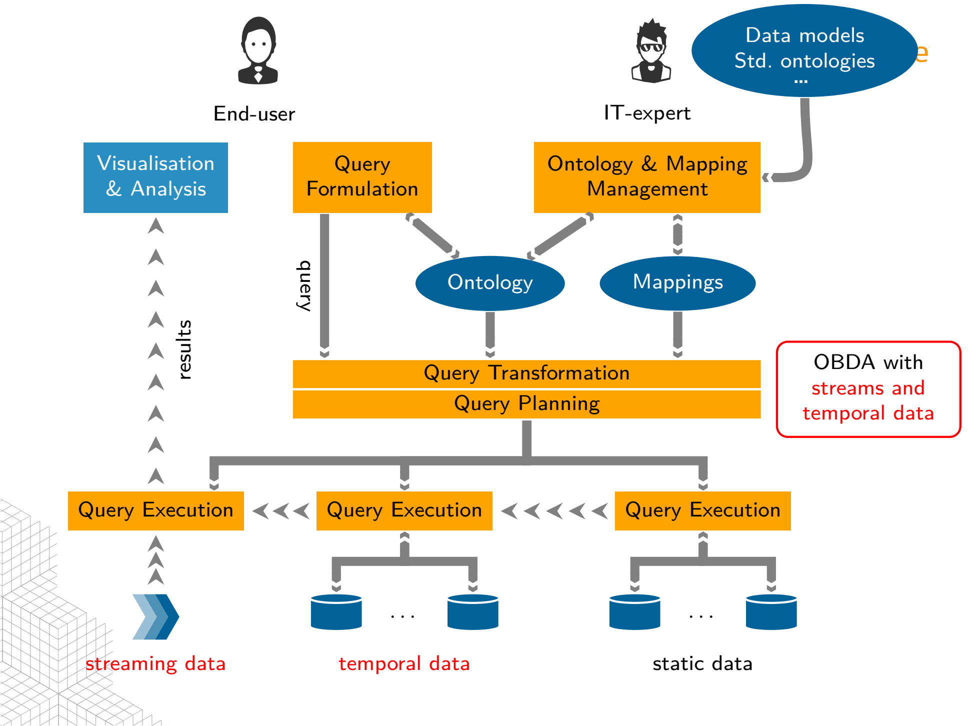
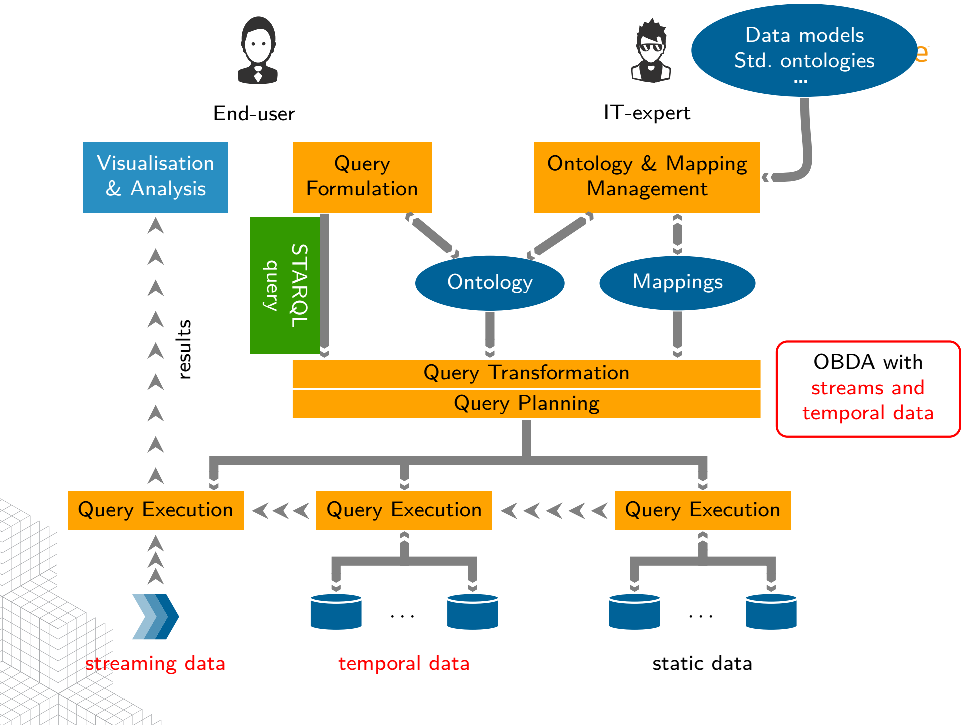
Stream-temporal Querying with Ontologies

Ralf Möller
Christian Neuenstadt
Özgür L. Özçep

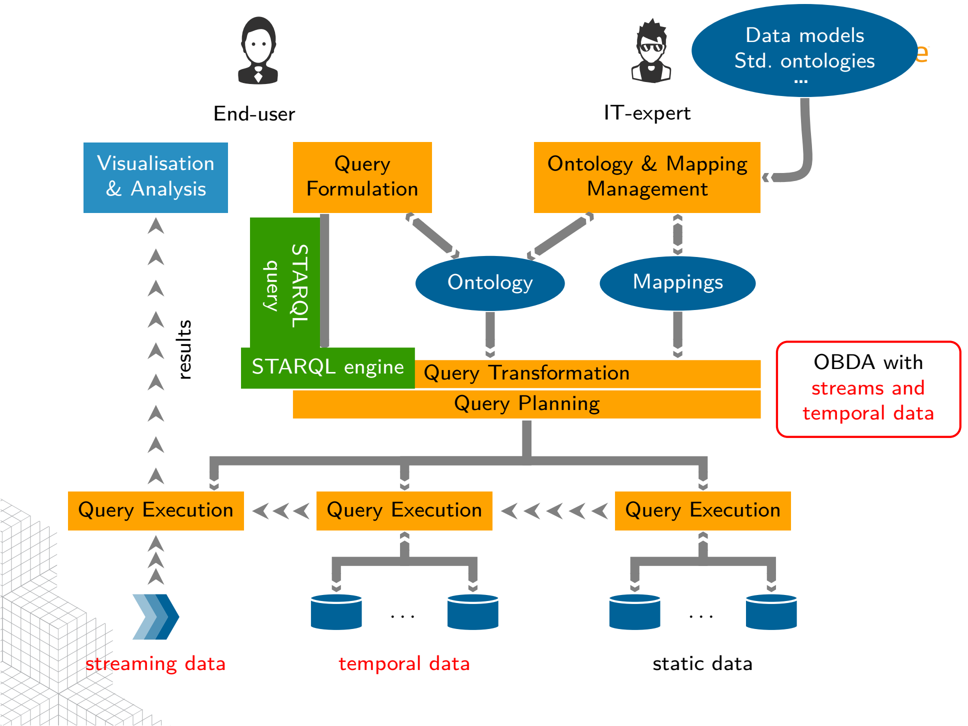
Hidest '15, Dresden



End-user
IT-expert
Data models
Std. ontologies
...
Visualisation
& Analysis
Query
Formulation
Ontology & Mapping
Management
query
results
Ontology
Mappings
Query Transformation
Query Planning
OBDA with
streams and
temporal data
Query Execution
Query Execution
Query Execution
. . .
. . .
streaming data
temporal data
static data



End-user
IT-expert
Data models
Std. ontologies
...
Visualisation & Analysis
Query Formulation
Ontology & Mapping Management
STARQL query
Ontology
Mappings
results
Query Transformation
Query Planning
OBDA with streams and temporal data
Query Execution
Query Execution
Query Execution
. . .
. . .
streaming data
temporal data
static data


Data models
Std. ontologies
...
End-user
IT-expert
Visualisation
& Analysis
Query
Formulation
Ontology & Mapping
Management
STARQL
query
Ontology
Mappings
results
STARQL engine
Query Transformation
Query Planning
OBDA with
streams and
temporal data
Query Execution
Query Execution
Query Execution
...
...
streaming data
temporal data
static data

Measurement Scenario

Optique

- Simplified Schema

```
CREATE TABLE measurement (Mtimestamp,SID,Mval);
CREATE TABLE sensor (SID,assemblypart,name)
```

- Datasets from Siemens Turbine Diagnostic Center

Dataset	Total No. Meas.	Timespan	Sensors	Meas. per Day/Sensor	Total Data Size
Ds1	82080	3 Days	19	1440	5 MB
Ds2	210,000	1506 Days	3	46.5	10 MB
Ds3	515,845,000	1824 Days	204	1386	23,000 MB

Window-Based Data Processing

Optique

Information Need

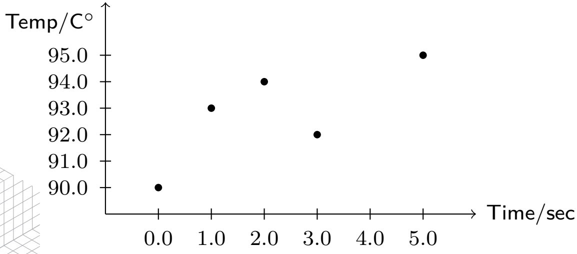
Starting with time point 00:00 on 1.1.2005, give me every second those temperature sensors whose value grew monotonically in the last 2 seconds.

- Information need for
  - historical data (temporal query) with period of interest
    $\Longrightarrow$ needed for reactive diagnosis
  - streamed data (continuous query)
    $\Longrightarrow$ needed for proactive diagnosis
- Goal: Use the same query language
  - For historical queries, windows can be processed in parallel with no real-time constraints

Example: Input Stream

Input: One stream S_{Msmt} of measurement assertions.

$$\begin{aligned} S_{Msmt} \;=\; \{\; & val(s_0, 90^\circ)\langle 0s\rangle, \\ & val(s_0, 93^\circ)\langle 1s\rangle, \\ & val(s_0, 94^\circ)\langle 2s\rangle, \\ & val(s_0, 92^\circ)\langle 3s\rangle, \\ & val(s_0, 95^\circ)\langle 5s\rangle \\ & \ldots \} \end{aligned}$$



Mappings

Optique

- Mapping historical data

$m1 : val(x, y)\langle z \rangle \quad \longleftarrow$

```
SELECT f(SID) AS x, Mval AS y, MtimeStamp AS z
FROM MEASUREMENT-TABLE
```

- Mapping streams

$m2 : val(x, y)\langle z \rangle \quad \longleftarrow$

```
SELECT Rstream(f(SID) AS x, Mval AS y,
               MtimeStamp AS z)
FROM MEASUREMENT-REL-STREAM
```

Representation in STARQL

Optique

```
PREFIX : <http://www.optique-project.eu/siemens>
CREATE STREAM S_out AS
CONSTRUCT  GRAPH NOW { ?s rdf:type MonInc }
FROM STREAM S_Msmt [NOW-2s, NOW]->"1S"^^xsd:duration
     WITH START = "2005-01-01T01:00:00CET"^^xsd:dateTime,
          END = "2005-01-01T02:00:00CET"^^xsd:dateTime
     STATIC ABOX <http://www.optique-project.eu/siemens/ABoxstatic>,
     TBOX <http://www.optique-project.eu/siemens/TBox>
USING PULSE WITH
            START = "2005-01-01T00:00:00CET"^^xsd:dateTime,
            FREQUENCY = "1S"^^xsd:duration
WHERE { ?s rdf:type :TempSens }
SEQUENCE BY StdSeq AS seq
HAVING FORALL ?i < ?j IN seq
  FOR ALL ?x,?y:
    IF (GRAPH ?i { ?s :val ?x }  AND GRAPH ?j { ?s  :val ?y })
    THEN ?x <= ?y
```

Output Specification

Optique

```
PREFIX : <http://www.optique-project.eu/siemens>
CREATE STREAM S_out AS
CONSTRUCT  GRAPH NOW { ?s rdf:type MonInc }
FROM STREAM S_Msmt [NOW-2s, NOW]->"1S"^^xsd:duration
     WITH START = "2005-01-01T01:00:00CET"^^xsd:dateTime,
          END = "2005-01-01T02:00:00CET"^^xsd:dateTime
     STATIC ABOX <http://www.optique-project.eu/siemens/ABoxstatic>,
     TBOX <http://www.optique-project.eu/siemens/TBox>
USING PULSE WITH
            START = "2005-01-01T00:00:00CET"^^xsd:dateTime,
            FREQUENCY = "1S"^^xsd:duration
WHERE { ?s rdf:type :TempSens }
SEQUENCE BY StdSeq AS seq
HAVING FORALL ?i < ?j IN seq
  FOR ALL ?x,?y:
    IF (GRAPH ?i { ?s :val ?x }  AND GRAPH ?j { ?s  :val ?y })
    THEN ?x <= ?y
```

Output Format and Frequency

Optique

$$
\begin{aligned}
S_{Msmt} \;=\; \{\; & val(s_0, 90^\circ)\langle 0s\rangle, \\
& val(s_0, 93^\circ)\langle 1s\rangle, \\
& val(s_0, 94^\circ)\langle 2s\rangle, \\
& val(s_0, 92^\circ)\langle 3s\rangle, \\
& \\
& val(s_0, 95^\circ)\langle 5s\rangle \\
& \ldots \}
\end{aligned}
$$

$$
\begin{aligned}
S_{out} \;=\; \{\; & MonInc(s_0)\langle 0s\rangle, \\
& MonInc(s_0)\langle 1s\rangle, \\
& MonInc(s_0)\langle 2s\rangle, \\
& \\
& \\
& MonInc(s_0)\langle 5s\rangle \\
& \ldots \}
\end{aligned}
$$

Streams and Sequencing

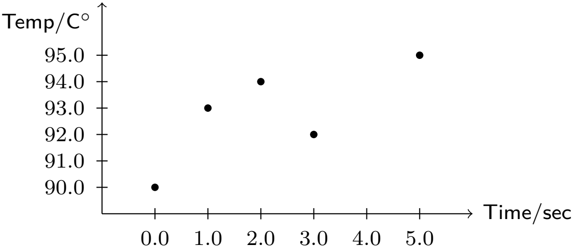
Optique

```
PREFIX : <http://www.optique-project.eu/siemens>
CREATE STREAM S_out AS
CONSTRUCT  GRAPH NOW { ?s rdf:type MonInc }
FROM STREAM S_Msmt [NOW-2s, NOW]->"1S"^^xsd:duration
       WITH START = "2005-01-01T01:00:00CET"^^xsd:dateTime,
            END = "2005-01-01T02:00:00CET"^^xsd:dateTime
       STATIC ABOX <http://www.optique-project.eu/siemens/ABoxstatic>,
       TBOX <http://www.optique-project.eu/siemens/TBox>
USING PULSE WITH
            START = "2005-01-01T00:00:00CET"^^xsd:dateTime,
            FREQUENCY = "1S"^^xsd:duration
WHERE { ?s rdf:type :TempSens }
SEQUENCE BY StdSeq AS seq
HAVING FORALL ?i < ?j IN seq
  FOR ALL ?x,?y:
    IF (GRAPH ?i { ?s :val ?x }  AND GRAPH ?j { ?s  :val ?y })
    THEN ?x <= ?y
```

Window Semantics

Optique

- `S_Msmt [NOW-2s,NOW]->1s`: stream of temporal ABoxes
- Sliding movement as in CQL but with timestamp preservation



Window sliding every second

Time	Window contents
$0s$	$val(s_0, 90°)\langle 0s\rangle$
$1s$	$val(s_0, 90°)\langle 0s\rangle$, $val(s_0, 93°)\langle 1s\rangle$
$2s$	$val(s_0, 90°)\langle 0s\rangle$, $val(s_0, 93°)\langle 1s\rangle$, $val(s_0, 94°)\langle 2s\rangle$
$3s$	$val(s_0, 93°)\langle 1s\rangle$, $val(s_0, 94°)\langle 2s\rangle$, $val(s_0, 92°)\langle 3s\rangle$
$4s$	$val(s_0, 94°)\langle 2s\rangle$, $val(s_0, 92°)\langle 3s\rangle$
$5s$	$val(s_0, 92°)\langle 3s\rangle$, $val(s_0, 95°)\langle 5s\rangle$

Timestamps to Sequences

Optique

Time	Window contents before sequencing	
...	...	
$5s$	$val(s_0, 92°)\langle 3s\rangle, \ val(s_0, 95°)\langle 5s\rangle$	
Time	Window contents after standard sequencing	$SEQ1$
...	...	
5s	$\{val(s_0, 92°)\}\langle 0\rangle, \ \{val(s_0, 95°)\}\langle 1\rangle$	{0,1}

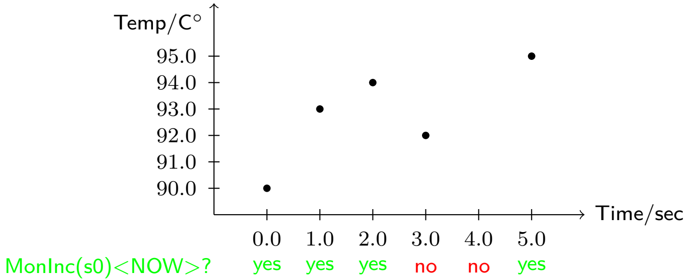
- Timestamped assertions are grouped to ABoxes (RDF graphs) with state index
- Information on timestamps and on their distance gets lost
- For transformation to SQL: the timestamp-index-window id has to be stored
  $\Longrightarrow$ window(Index,Timestamp,WID)

WHERE and HAVING

Optique

```
PREFIX : <http://www.optique-project.eu/siemens>
CREATE STREAM S_out AS
CONSTRUCT  GRAPH NOW { ?s rdf:type MonInc }
FROM STREAM S_Msmt [NOW-2s, NOW]->"1S"^^xsd:duration
     WITH START = "2005-01-01T01:00:00CET"^^xsd:dateTime,
          END = "2005-01-01T02:00:00CET"^^xsd:dateTime
      STATIC ABOX <http://www.optique-project.eu/siemens/ABoxstatic>,
      TBOX <http://www.optique-project.eu/siemens/TBox>
USING PULSE WITH
            START = "2005-01-01T00:00:00CET"^^xsd:dateTime,
            FREQUENCY = "1S"^^xsd:duration
WHERE { ?s rdf:type :TempSens }
SEQUENCE BY StdSeq AS seq
HAVING FORALL ?i < ?j IN seq
  FOR ALL ?x,?y:
    IF (GRAPH ?i { ?s :val ?x }  AND GRAPH ?j { ?s  :val ?y })
    THEN ?x <= ?y
```

Testing the Conditions

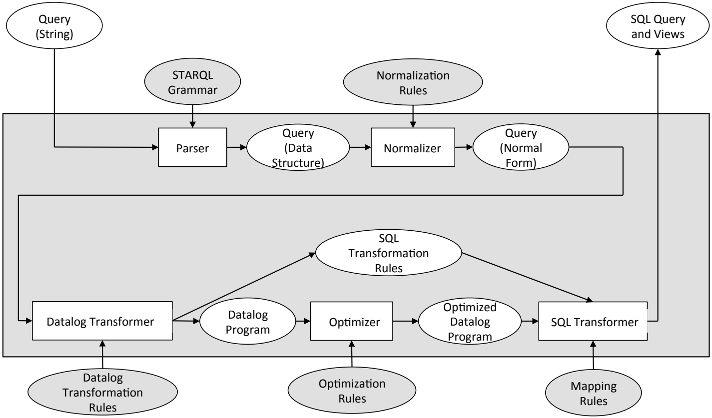


$$
\begin{aligned}
S_{Msmt} = \{ & val(s_0, 90^\circ C)\langle 0s\rangle, \\
& val(s_0, 93^\circ C)\langle 1s\rangle, \\
& val(s_0, 94^\circ C)\langle 2s\rangle, \\
& val(s_0, 92^\circ C)\langle 3s\rangle, \\
& val(s_0, 95^\circ C)\langle 5s\rangle \\
& \ldots \}
\end{aligned}
$$

$$
\begin{aligned}
S_{out} = \{ & MonInc(s_0)\langle 0s\rangle, \\
& MonInc(s_0)\langle 1s\rangle, \\
& MonInc(s_0)\langle 2s\rangle, \\
& \\
& MonInc(s_0)\langle 5s\rangle \\
& \ldots \}
\end{aligned}
$$

STARQL Engine: Architecture

Optique



STARQL Engine: Evaluation

Optique

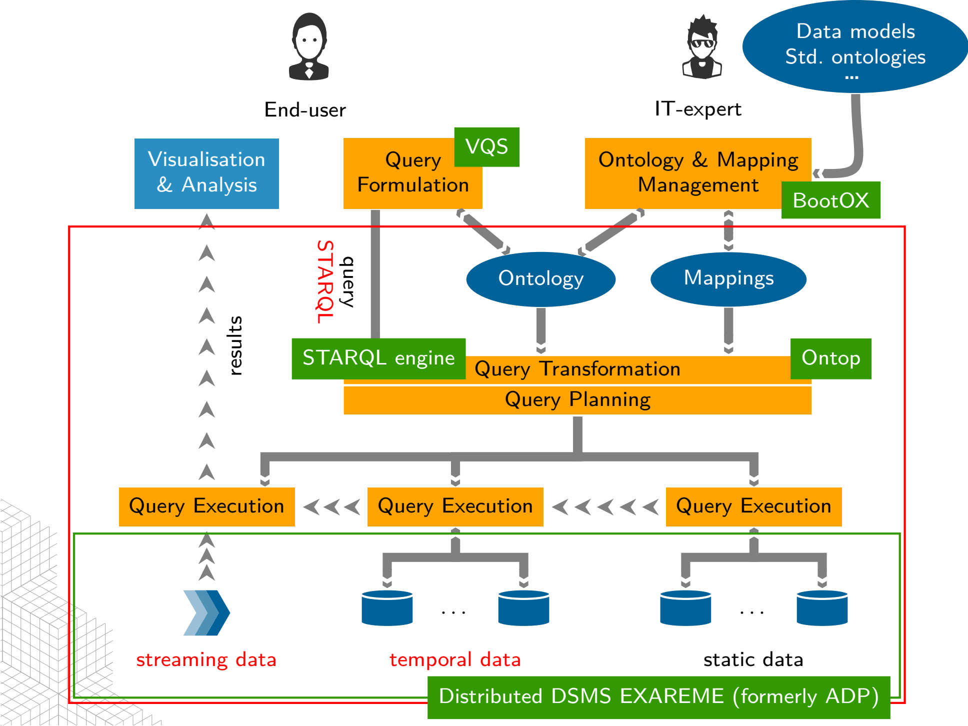
- Tested feasability of engine
- Datasets of increasing size (5MB, 10MB, 23,000MB)
- Two queries: monotonic increase (quadratic) and threshold query (linear)
- Two different approaches
  - preprocessing of window table (temporal querying)
  - incremental generation of window table (temporal and stream)
- Results

  Incremental approach more feasible; but preprocessing step useful for caching.

OPTIQUE

Optique

- EU 7th framework program
- Two big data use cases from industrial partners
  - STATOIL SAS: Querying data on wellbore related DBs
  - SIEMENS: Querying sensor and event data from (gas) turbines
- Optique platform: OBDA + User Friendliness + Scalable Rewriting + Elastic Clouds + Real-Time Processing
- For more information: http://www.optique-project.eu/
- Demos and Optique assets to be found at fluidOps http://appcenter.fluidops.net



End-user
IT-expert
Data models
Std. ontologies
...
Visualisation
& Analysis
Query
Formulation
VQS
Ontology & Mapping
Management
BootOX
STARQL
query
results
Ontology
Mappings
STARQL engine
Query Transformation
Ontop
Query Planning
Query Execution
Query Execution
Query Execution
streaming data
temporal data
static data
Distributed DSMS EXAREME (formerly ADP)

Take-Home Message

Optique

One can build stream-temporal OBDA engines that

1. have a query language with neat formal semantics,
2. use standard backend systems (such as PostgreSQL)
3. and provide good performance (under optimized transformation)