Towards Cascading Reasoning for Generic Edge Processing

Pieter Bonte, Femke Ongenae

GHENT
UNIVERSITY

Limitations of Cloud Computing for the IoT

Cloud

Processing

Environmental impact

High latency

Edge

Sensor Gateway

Sensor Gateway

Sensor Gateway

Sensor Gateway

Network congestion

Privacy concerns

Streams

High Velocity

"Real-time data"

CO2

Temperature

Loudness

IoT Devices
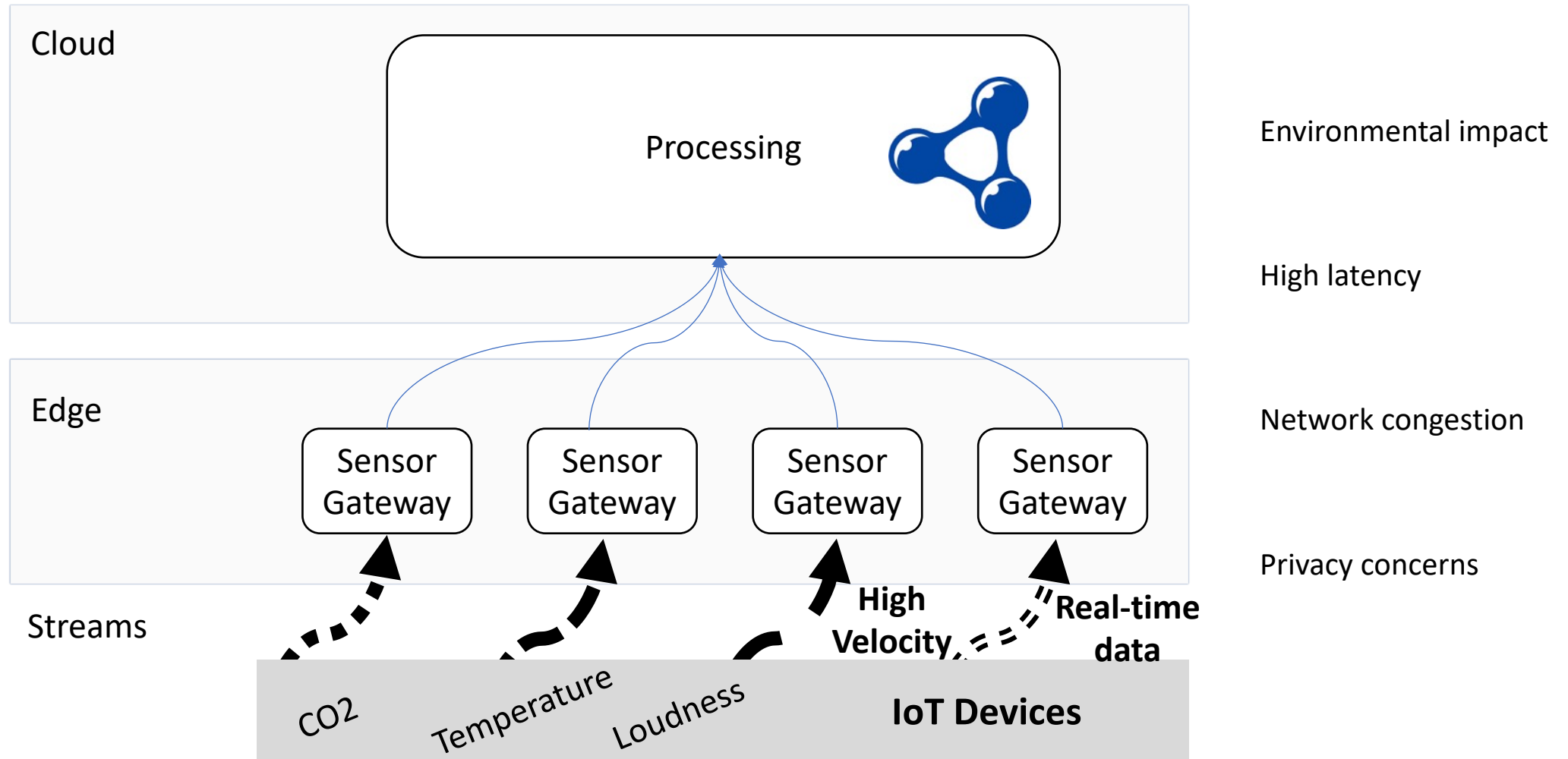
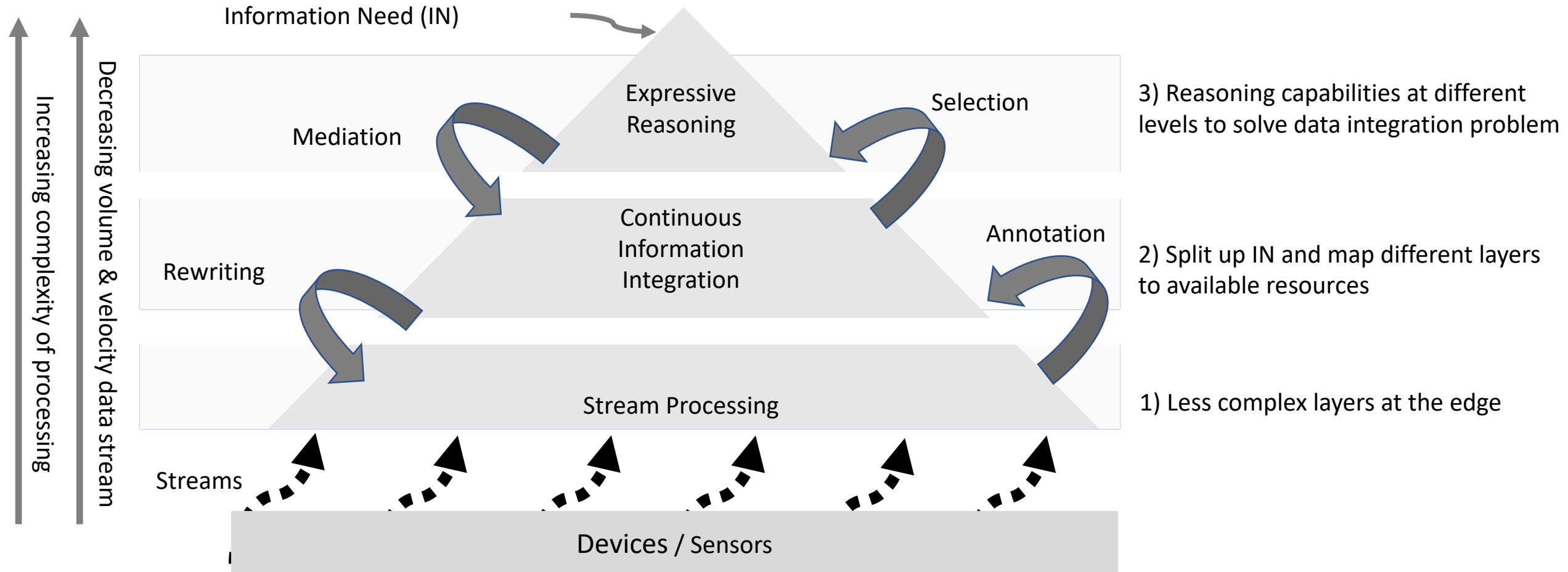# Potential of Edge and Fog Computing for the IoT

Cloud

Processing

Fog

Processing

Processing
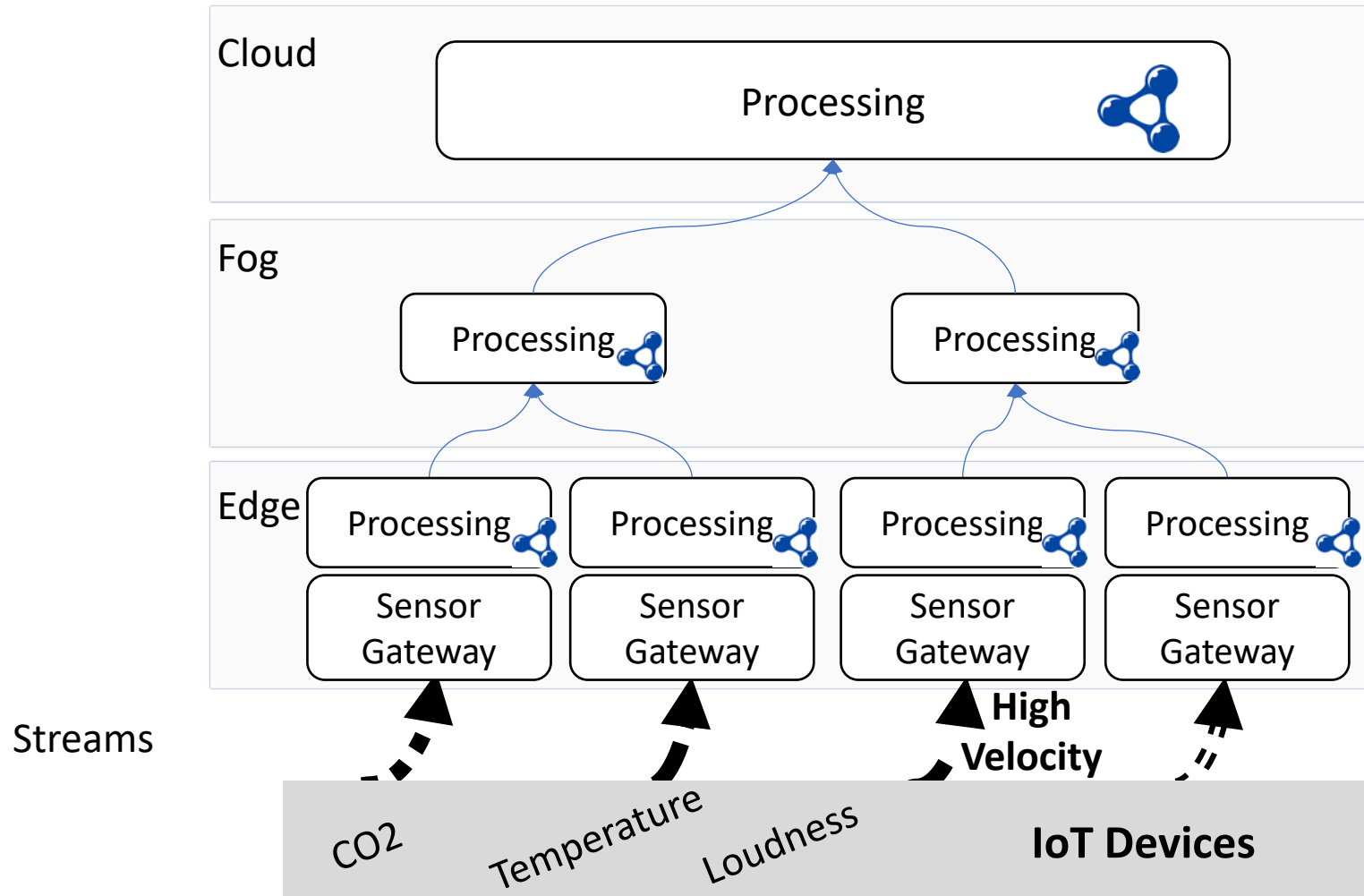
3) Data integration problem

2) Limited resources, heterogeneity, scale and complexity of the infrastructure

Edge

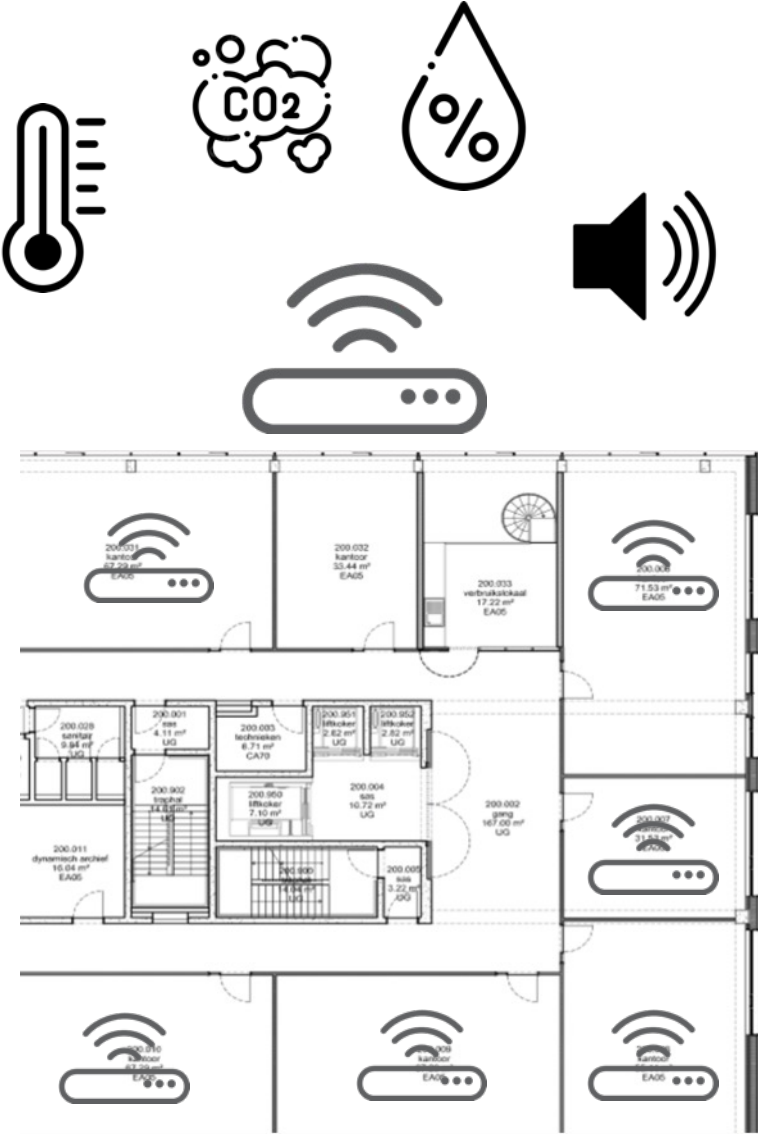Processing

Processing

Processing

Processing

Sensor Gateway

Sensor Gateway

Sensor Gateway

Sensor Gateway

Streams

High Velocity

1) More data is being produced than can be meaningfully processed

CO_2

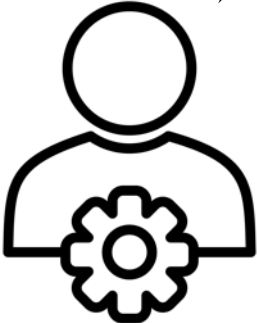Temperature

Loudness

IoT Devices

Cascading Reasoning

How can we rewrite queries and optimize the intermediate reasoning steps to offload them to the Edge and filter data early on?
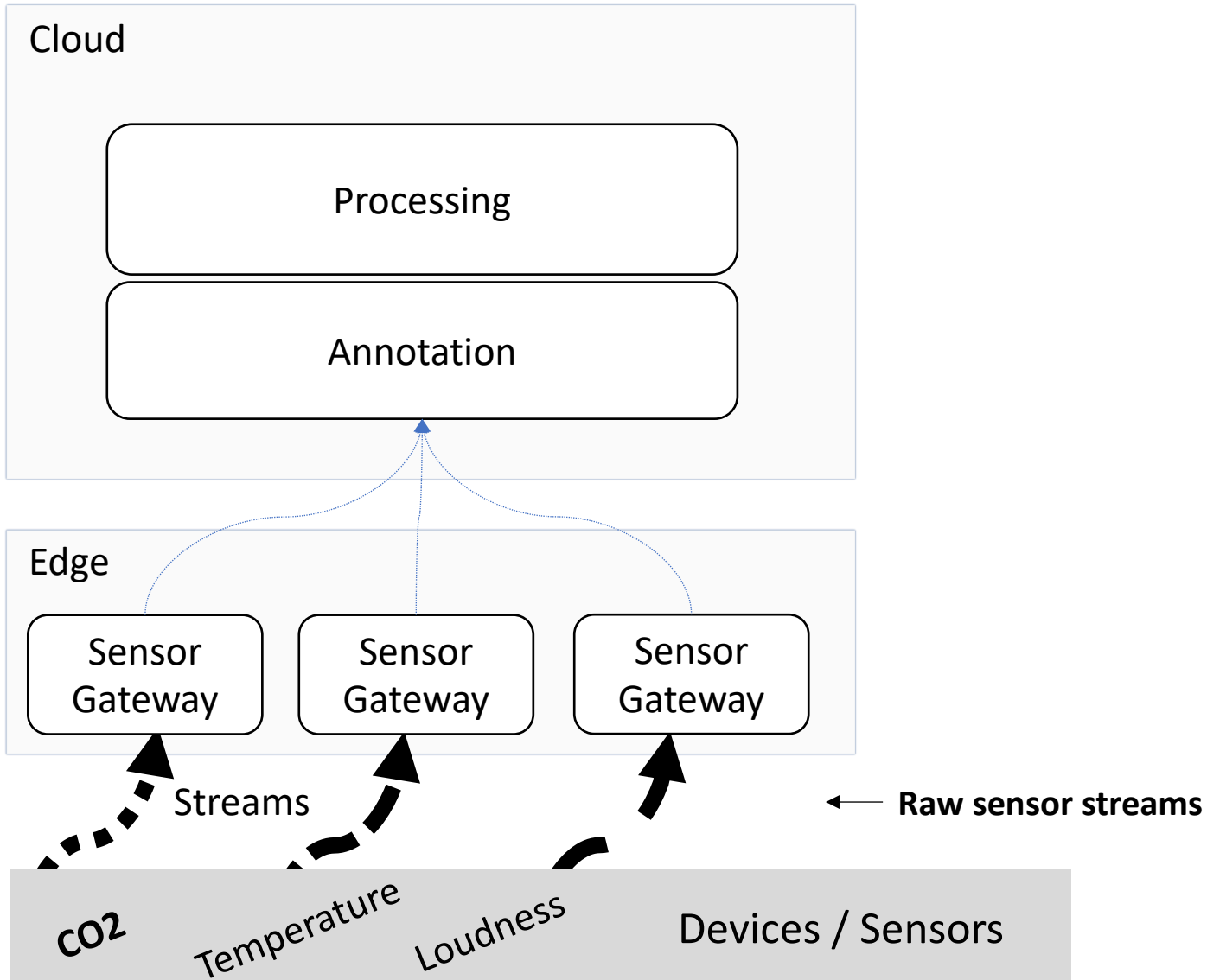
Running Example

Running Example

Cloud

Processing

Annotation

Edge

Sensor Gateway

Sensor Gateway

Sensor Gateway

Streams

← **Raw sensor streams**

CO2

Temperature

Loudness

Devices / Sensors

{ "sensorID": "1234",
  "observationID": "5674",
  "value": 765 }
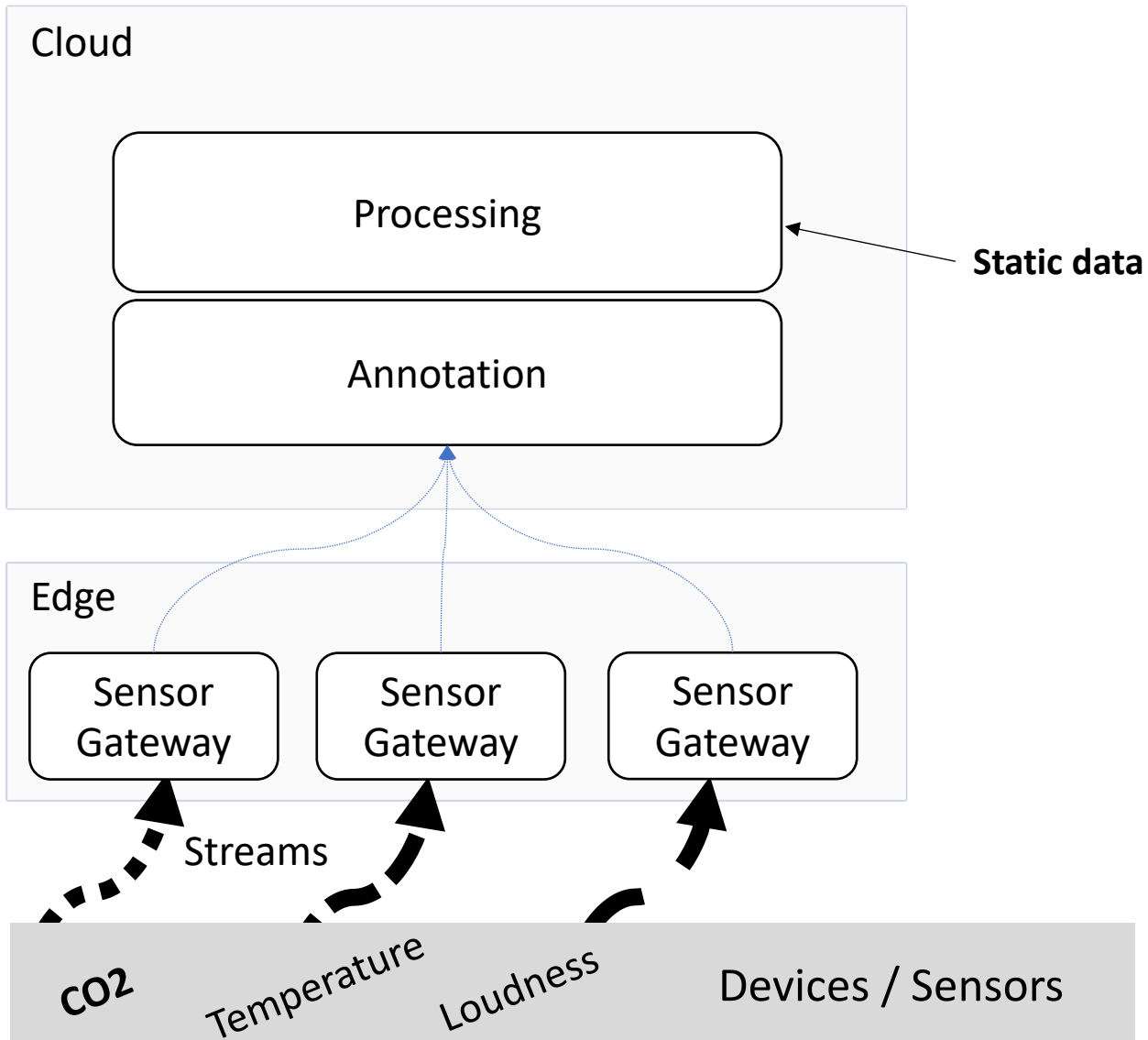
Running Example
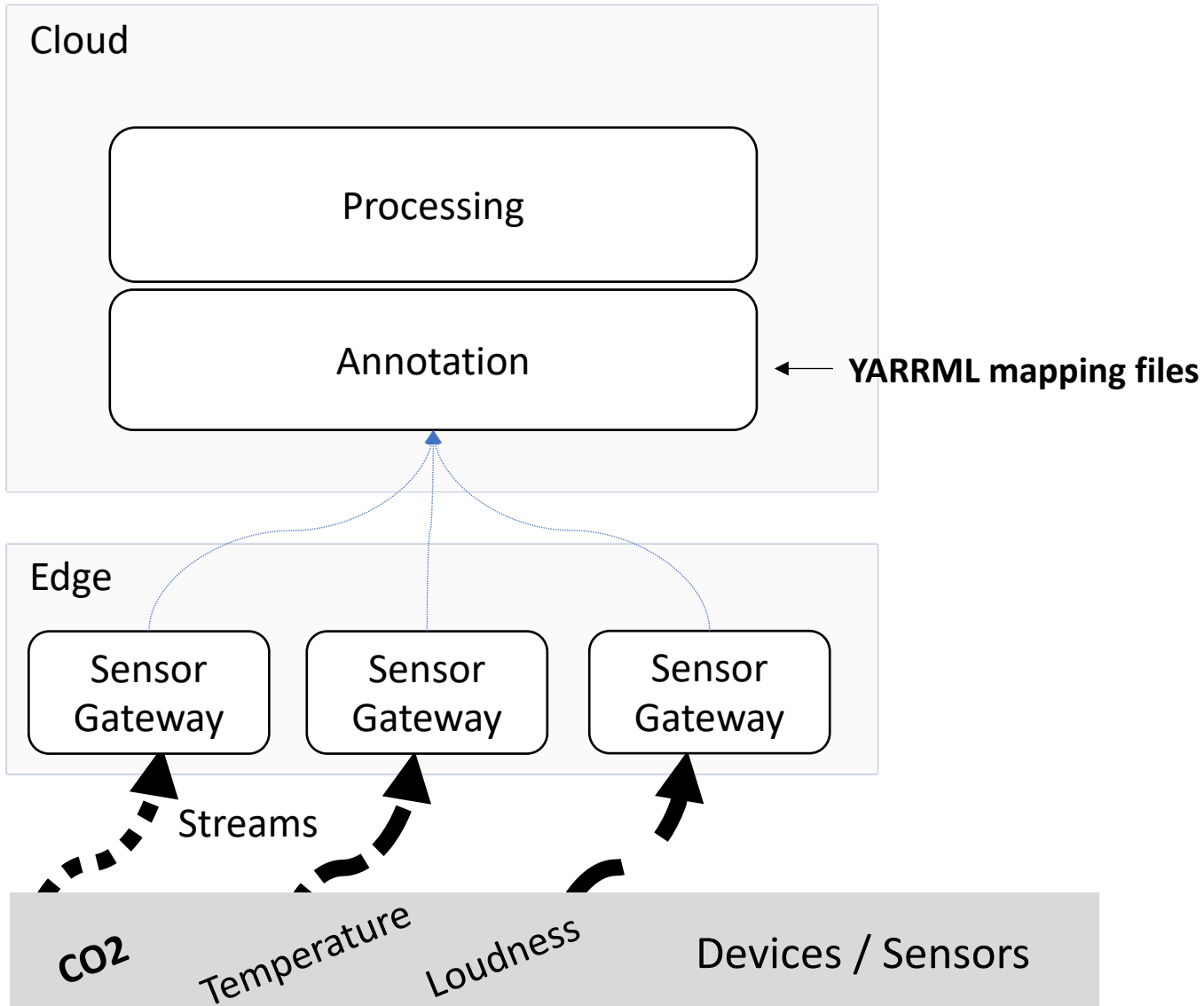

```
mappings:
  sensor:
    sources:
    s: iot:$(observationID)
    po:
      - [a, ssn:Observation]
      - [ssn:madeBySensor, iot:$(sensorID)]
      - [sosa:hasSimpleResult, $(value)]
```

Cloud

Processing

Annotation ← **YARRML mapping files**

Edge

Sensor Gateway
Sensor Gateway
Sensor Gateway

Streams

CO2 Temperature Loudness Devices / Sensors

Running Example

Cloud

Processing

← **Static data**

Annotation

Edge

Sensor Gateway | Sensor Gateway | Sensor Gateway

Streams

CO2   Temperature   Loudness   Devices / Sensors

...
:sensorX a sosa:Sensor;
     sosa:observes :temp, :loudness, :co2, humidity;
     :hasLocation :officeY.
:officeY :connectedTo :officeZ, officeQ;
     :hasName "200.009"^^xsd:string.
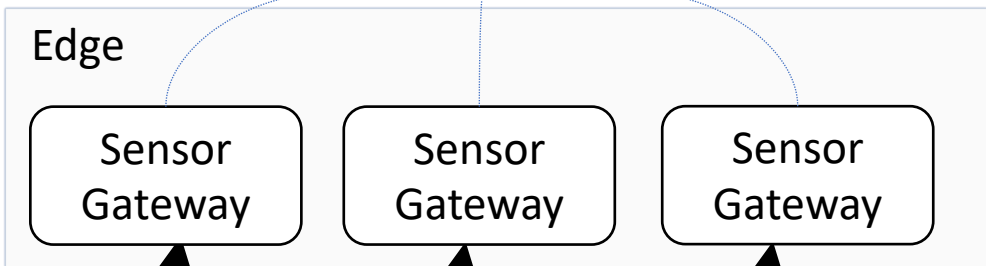...

Running Example
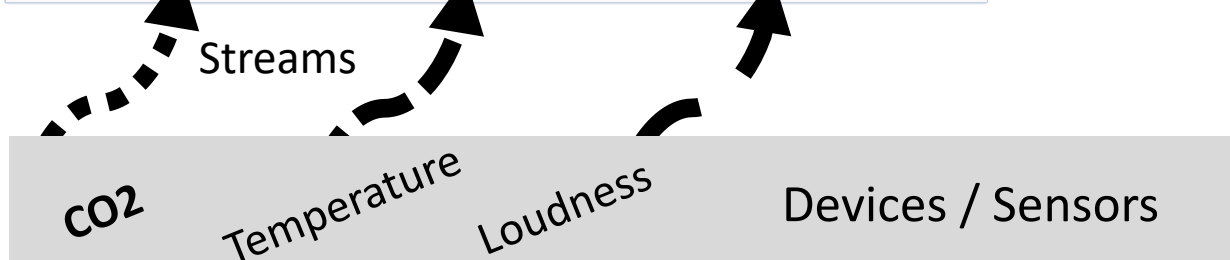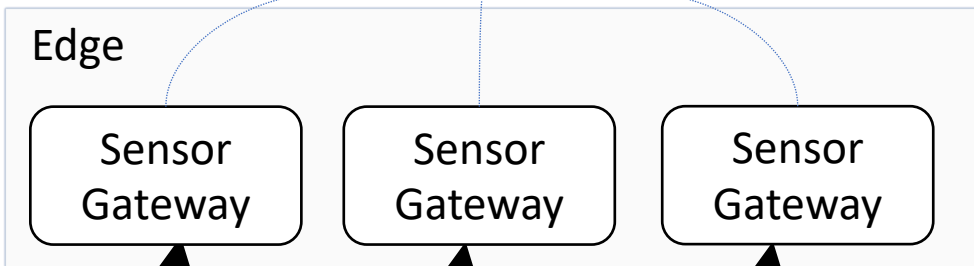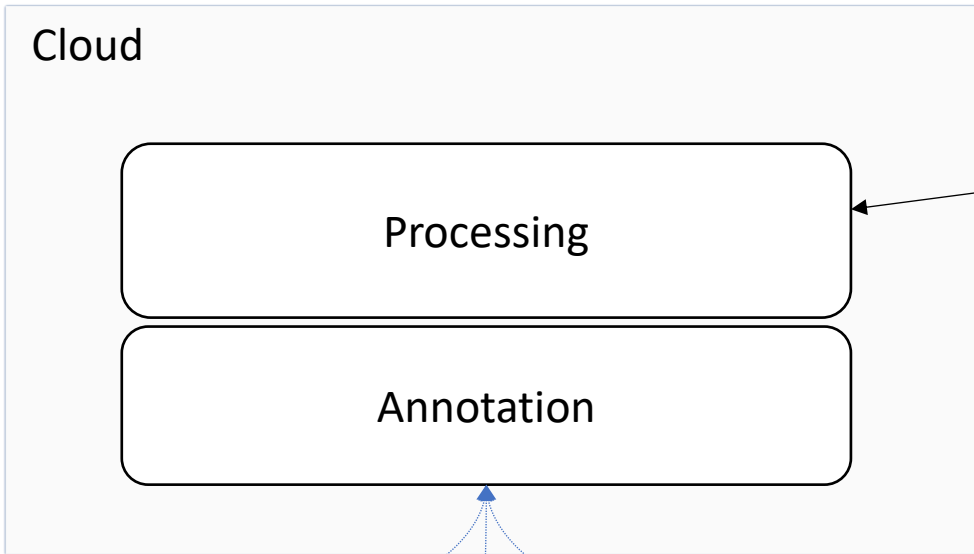
Rules

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}
R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
      => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
      => {?x a :LoudnessObservation}
R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature}
      => {?s a :TempSensor}
R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness}
      => {?s a :LoudnessSensor}

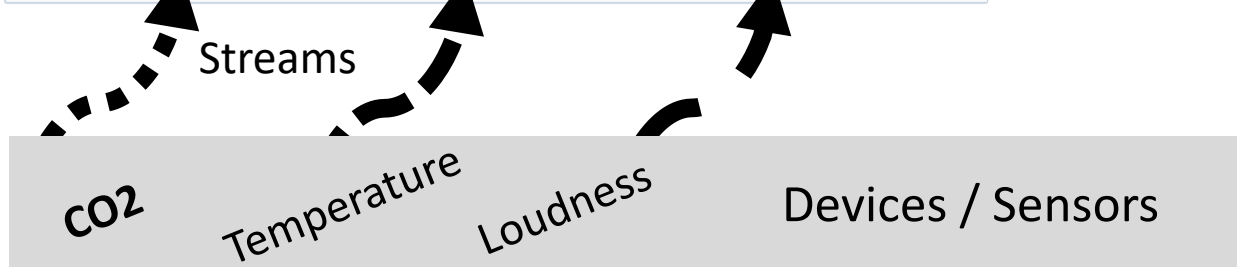# Running Example
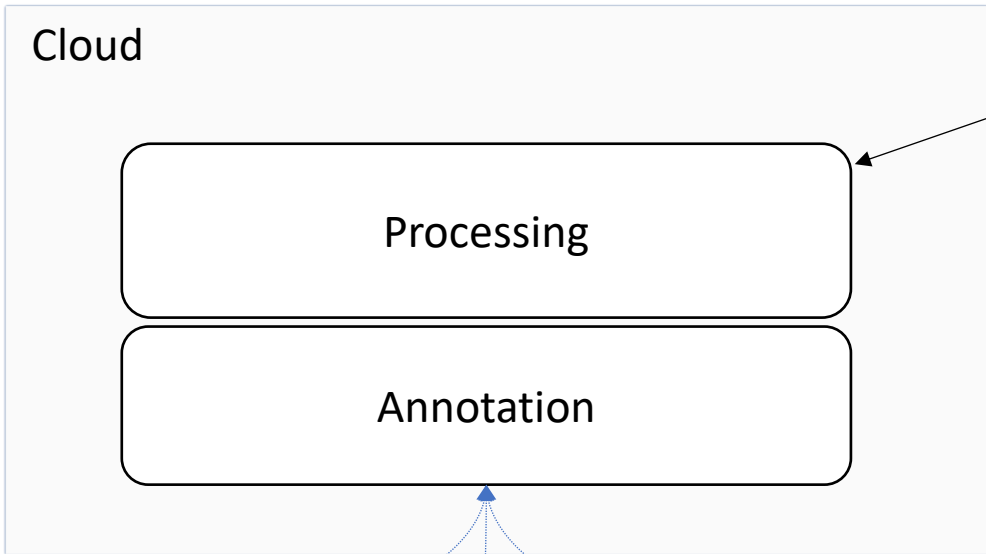
Queries

Q1:
```
?obs a Observation;           // stream
    hasSimpleResult ?value;   // stream
    madeBySensor ?sensor.     // stream
?sensor hasLocation ?loc.     // static
?loc hasName "200.009"        // static
```

Q2:
```
?obs a ComfortObservation;    // stream + domain knowledge
    hasSimpleResult ?value.   // stream
```
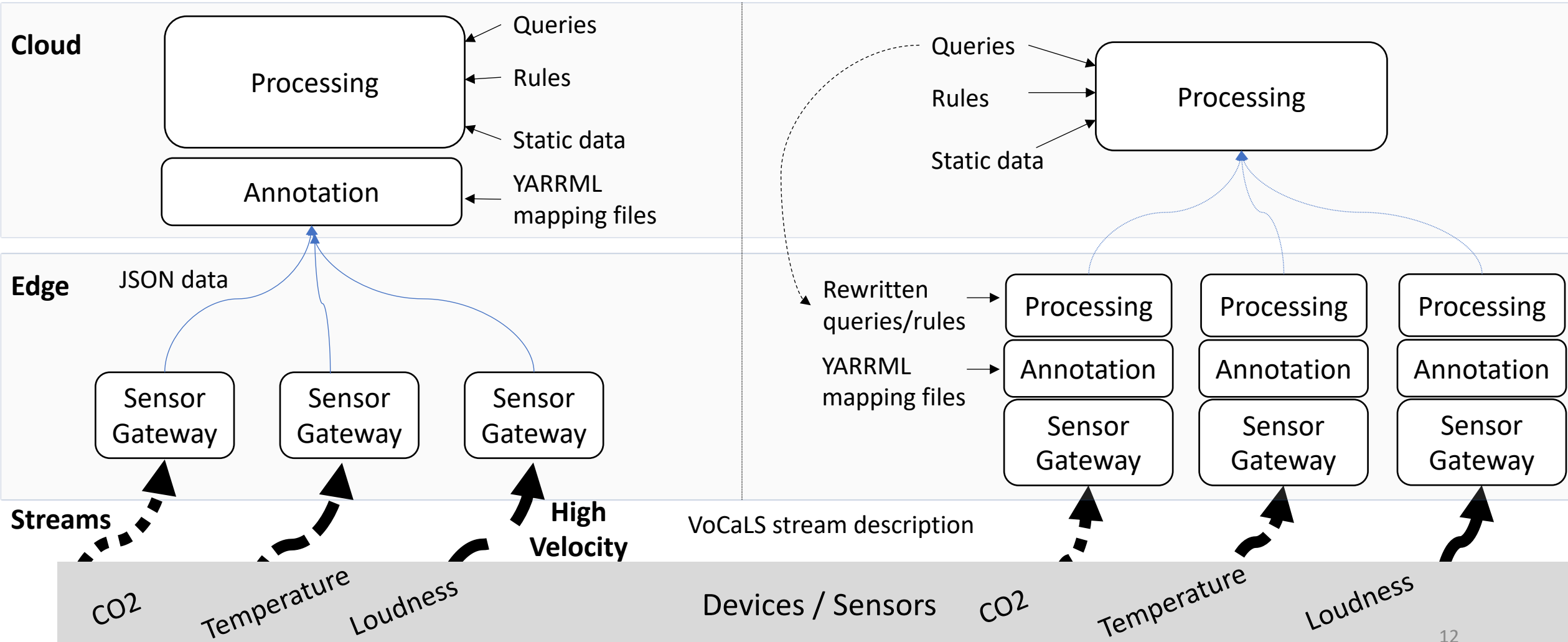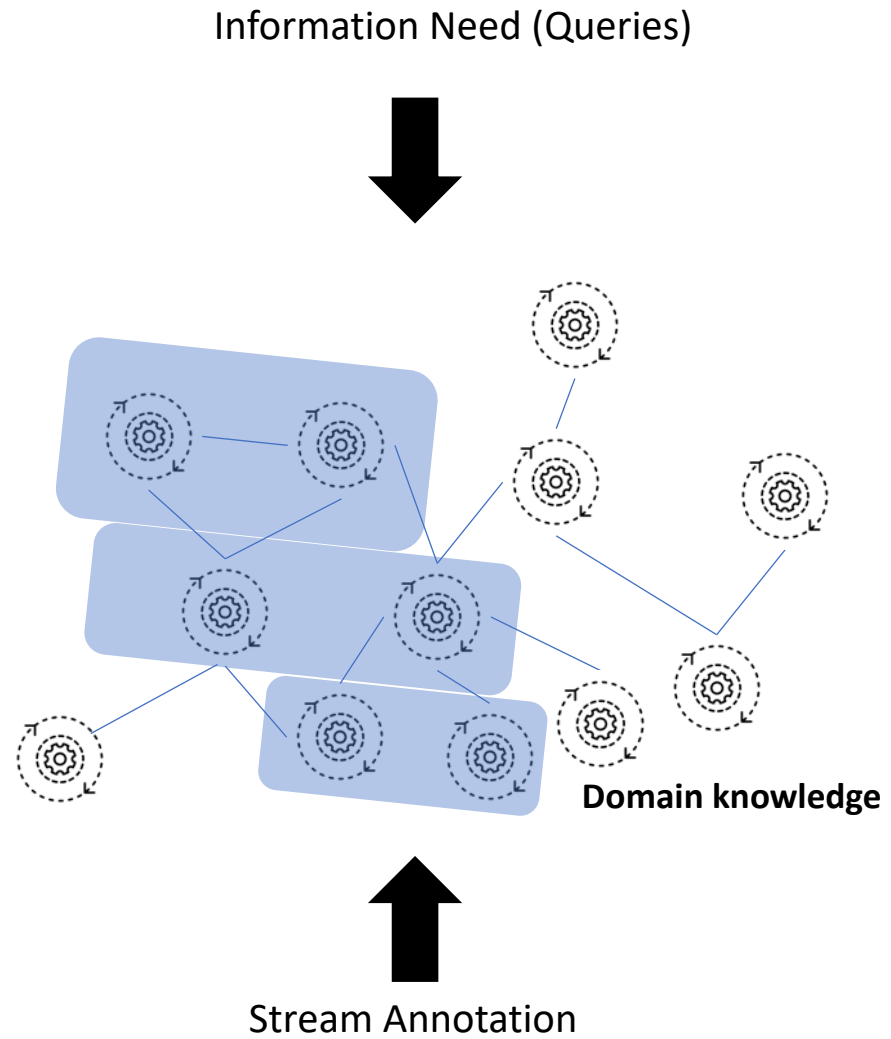
Cloud

Processing

Annotation

Edge

Sensor Gateway

Sensor Gateway

Sensor Gateway

Streams

CO2   Temperature   Loudness   Devices / Sensors

11

How do we get there?

a) Cloud paradigm

Cloud

Processing ← Queries

← Rules

← Static data

Annotation ← YARRML mapping files

Edge

JSON data

Sensor Gateway

Sensor Gateway

Sensor Gateway

Streams

High Velocity

CO2   Temperature   Loudness

b) Edge paradigm

Queries →

Rules →

Static data →

Processing

Rewritten queries/rules →

Processing

Annotation

Sensor Gateway

YARRML mapping files →

Processing

Annotation

Sensor Gateway

Processing

Annotation

Sensor Gateway

VoCaLS stream description

Devices / Sensors   CO2   Temperature   Loudness

How do we get there?

Information Need (Queries)

Domain knowledge

Stream Annotation

Shape Extraction

{?x a NodeShape. ?x targetClass ?clss} => {_:t a ?clss}
{?x a NodeShape. ?x sh:property [sh:path ?prop]}
=> {_:s ?prop _:o}

...

Annotation

RML2SHACL

SHACL

N3 Reasoning

Event BluePrint

```
mappings:
 sensor:
  sources:
  s: iot:$(observationID)
  po:
    - [a, ssn:Observation]
    - [ssn:madeBySensor, iot:$(sensorID)]
    - [sosa:hasSimpleResult, $(value)]
```

```
_:t  a ssn:Observation .
_:t ssn:madeBySensor _:q .
_:t sosa:hasSimpleResult _:z .
_:t :observedProperty _:p .
```

Query Rewriting

Rewritten Query:
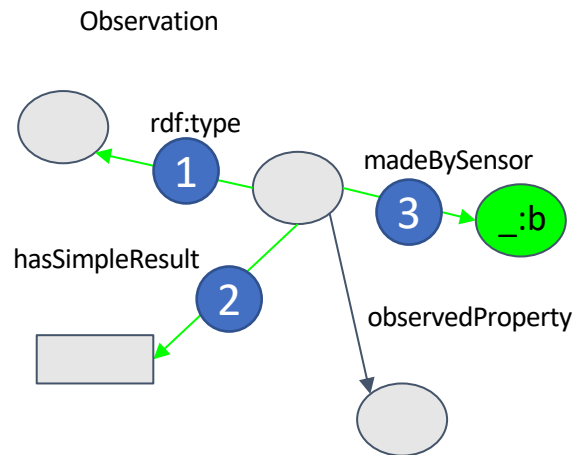
① ?obs a Observation;
②   hasSimpleResult ?value;
③   madeBySensor ?sensor.
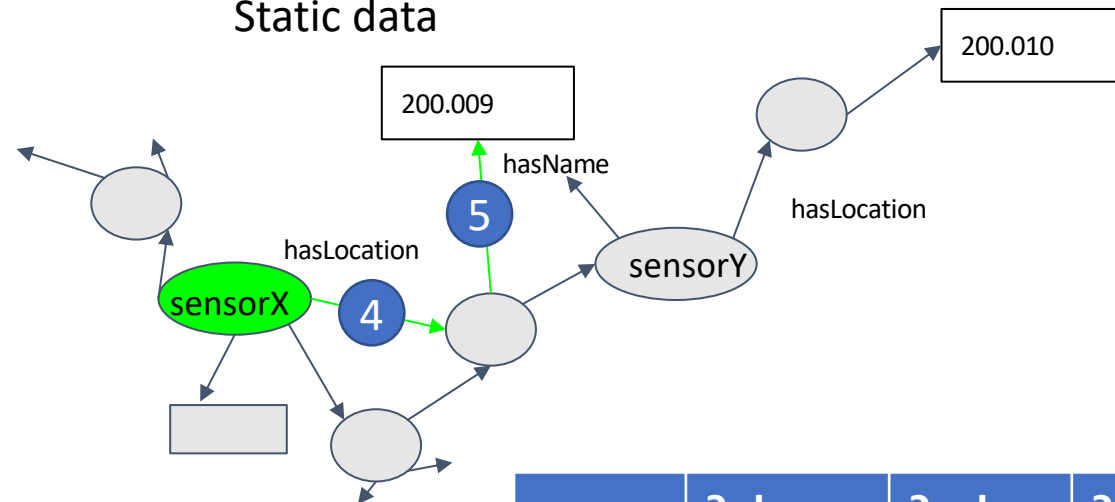④ ?sensor hasLocation ?loc.
⑤ ?loc hasName "200.009"

?obs a Observation;   // stream
   hasSimpleResult ?value; // stream
   madeBySensor **:sensorX**.   // stream

Event BluePrint

Static data

	?obs	?value	?sensor	?loc
value	_:t	_:z	:sensorX	:officeY
origin	Event	Event	**Event/Static**	static

Reasoning-enabled Rewriting

Query

   ?obs a :ComfortObservation;
         :hasSimpleResult ?val.

```
                              Comfort
                            Observation
```

Rules

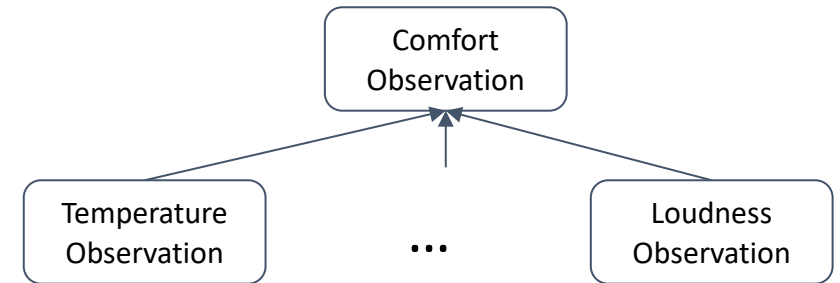   R1: {?x a :TempObservation} => {?x a :ComfortObservation}
   R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

   R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor} => {?x a :TempObservation}
   R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor} => {?x a :LoudnessObservation}
   R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature} => {?s a :TempSensor}
   R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}

Temperature Observation   ...   Loudness Observation

Reasoning-enabled Rewriting

R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature} => {?s a :TempSensor}

Query

R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
    => {?x a :TempObservation}

?obs a :**ComfortObservation**;
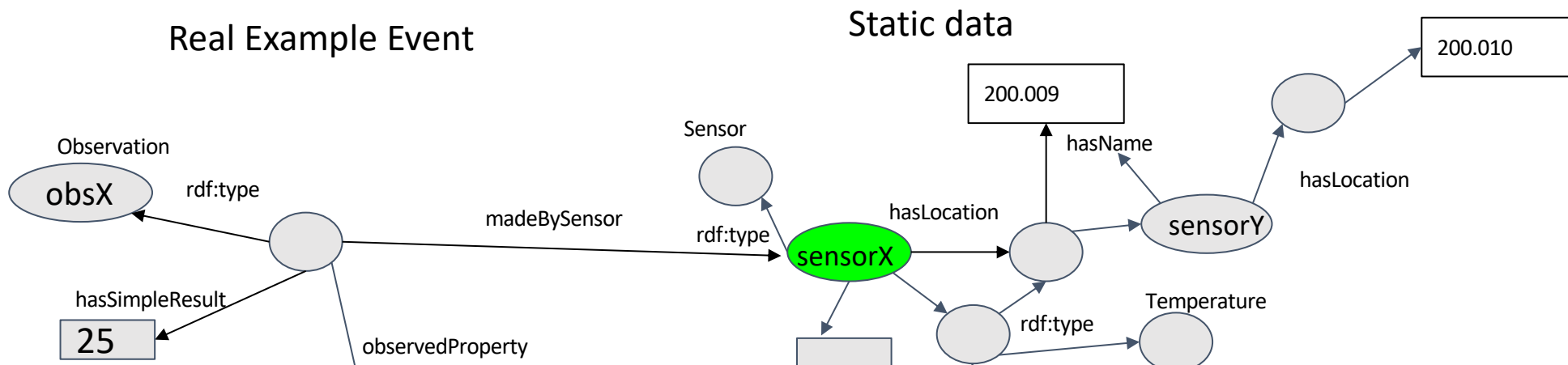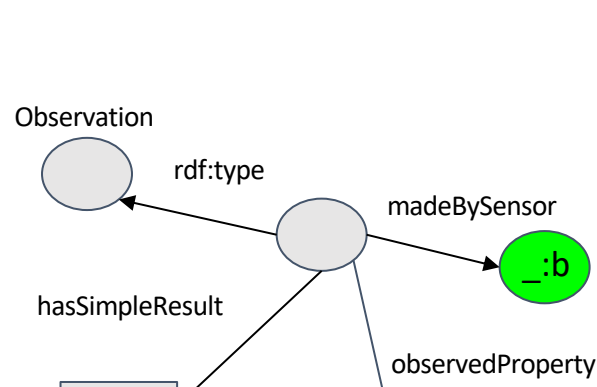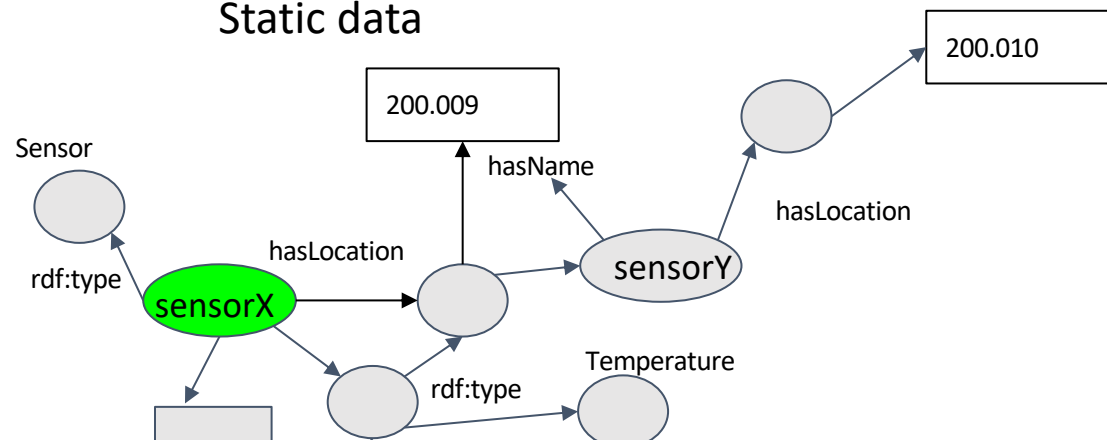   :hasSimpleResult ?val.

R1: {?x a :TempObservation} => {?x a :ComfortObservation}

Inferred:

:sensorX a :TempSensor

:obsX a :TempObservation

:obsX a :ComfortObservation

Real Example Event

Static data

Reasoning-enabled Rewriting

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
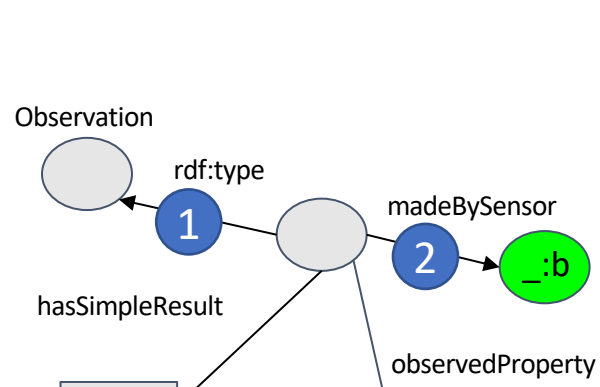**R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}**

R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
   => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
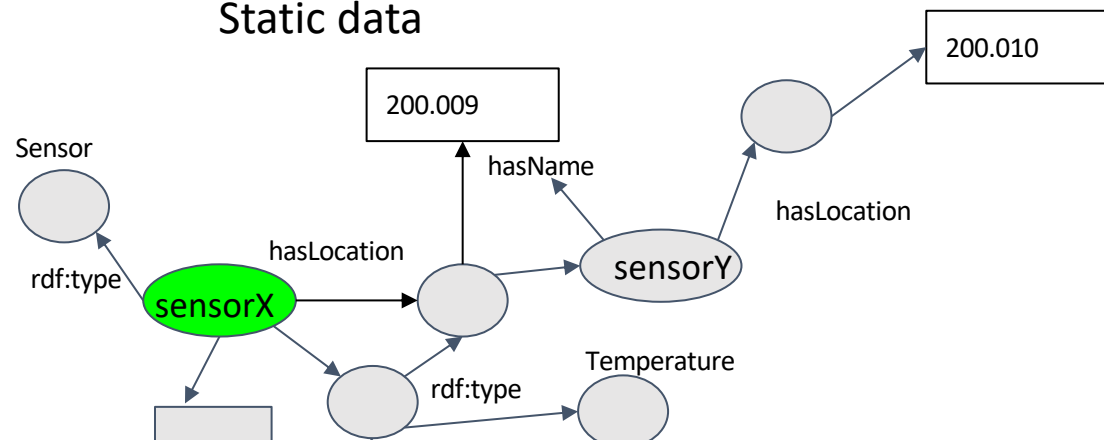   => {?x a :LoudnessObservation}
R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature} => {?s a :TempSensor}
R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}

Query

   ?obs a :**ComfortObservation**;
        :hasSimpleResult ?val.

Event BluePrint

Static data

18

Reasoning-enabled Rewriting

Query

    ?obs a :ComfortObservation;
            :hasSimpleResult ?val.

R1: {**?x a :TempObservation**} => {?x a :ComfortObservation}
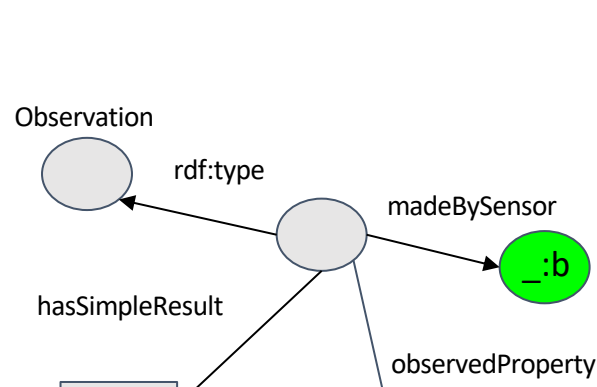R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

R3: {?x a :Observation, ?x :madeBySensor ?s, ?s a :TempSensor}
  => {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
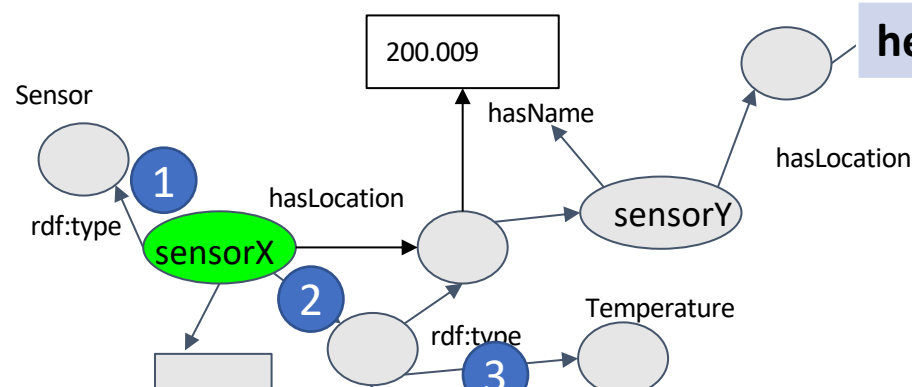 => {?x a :LoudnessObservation}
R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature} => {?s a :TempSensor}
R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}

Event BluePrint

Static data

Reasoning-enabled Rewriting

R1: {?x a :TempObservation} => {?x a :ComfortObservation}
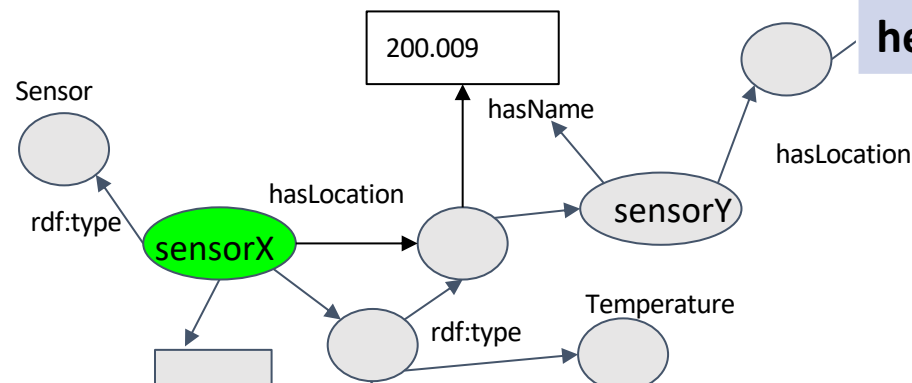R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

Query

?obs a :ComfortObservation;
:hasSimpleResult ?val.

R3: {?x a :Observation, ?x :madeBySensor ?s, **?s a :TempSensor}**
=> {?x a :TempObservation}
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
=> {?x a :LoudnessObservation}

(1)   (2)   (3)

R5: {?s a :Sensor, ?s :observes ?p, ?p a :Temperature} => {?s a :TempSensor}
R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}

	?s	?p
value	:sensorX	:propY
origin	Static	Static
head	**Yes**	No

Event BluePrint

Observation

rdf:type

madeBySensor

_:b

hasSimpleResult

observedProperty

Static data

Sensor

200.009

hasName

hasLocation

rdf:type

sensorY

(1)

hasLocation

sensorX

(2)

Temperature

rdf:type

(3)

Reasoning-enabled Rewriting

R1: {**?x a :TempObservation**} => {?x a :ComfortObservation}
R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

Query

    ?obs a :ComfortObservation;
              :hasSimpleResult ?val.

R3': {?x a :Observation, ?x :madeBySensor :sensorX}
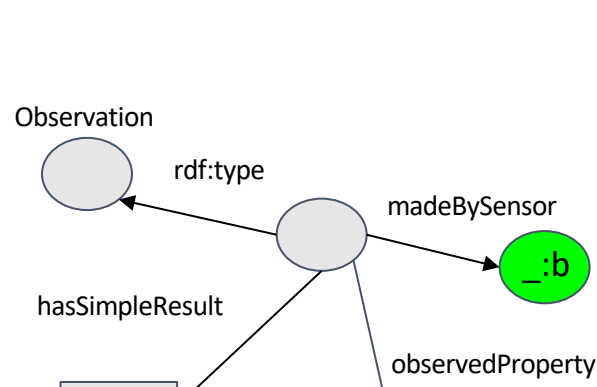 **=> {?x a :TempObservation}**
R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
  => {?x a :LoudnessObservation}
R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}
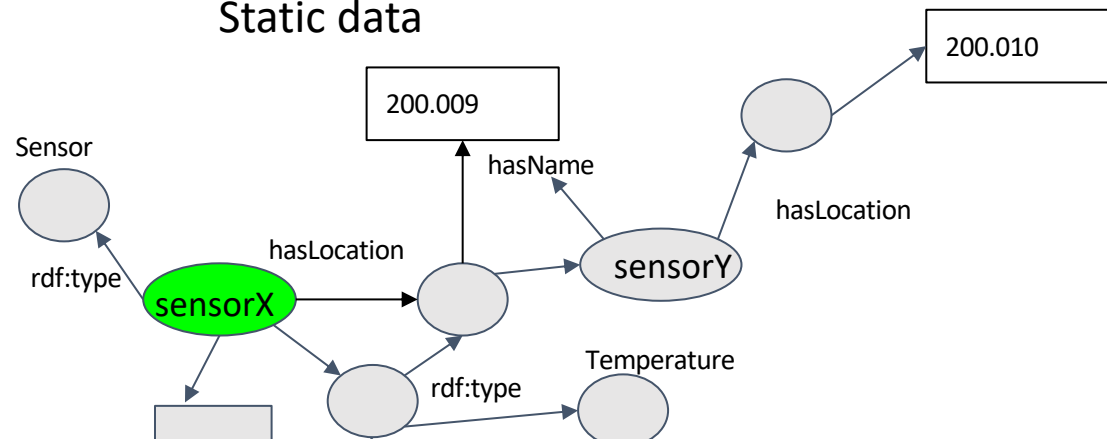
	?x
value	_:t
origin	Stream
head	**Yes**

Event BluePrint

Static data

Reasoning-enabled Rewriting

Identified Rules

R1: {?x a :TempObservation} => {?x a :ComfortObservation}

R2: {?x a :LoudnessObservation} => {?x a :ComfortObservation}

R3': {?x a :Observation, ?x :madeBySensor :sensorX}
   => {?x a :TempObservation}

R4: {?x a :Observation, ?x :madeBySensor ?s, ?s a :LoudnessSensor}
   => {?x a :LoudnessObservation}

R6: {?s a :Sensor, ?s :observes ?p, ?p a :Loudness} => {?s a :LoudnessSensor}

Query

?obs a :ComfortObservation;
        :hasSimpleResult ?val.

Event BluePrint

Static data

Reasoning-enabled Rewriting

Identified Rules

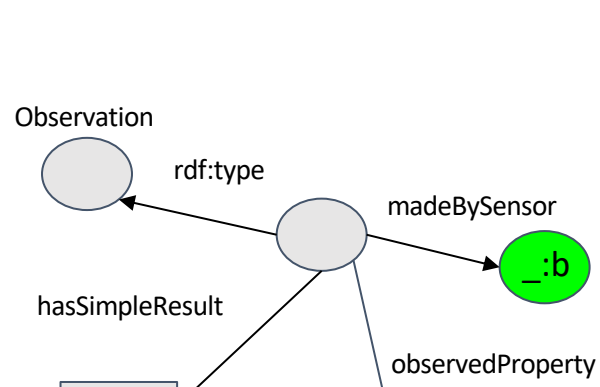R1: {?x a :TempObservation} => {?x a :ComfortObservation}

R3': {?x a :Observation, ?x :madeBySensor :sensorX}
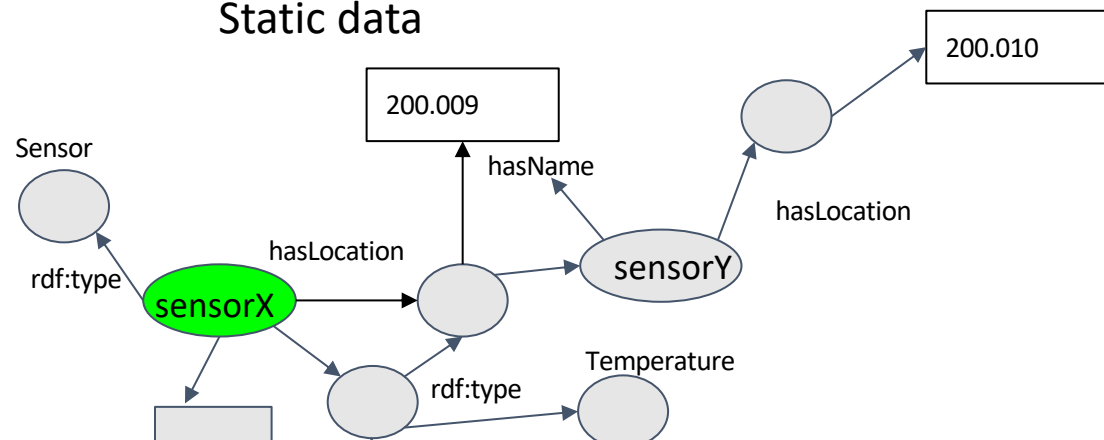   => {?x a :TempObservation}

Prune hierarchies

R3'': {?x a :Observation, ?x :madeBySensor :sensorX}
    => {?x a :ComfortObservation}

Query

?obs a :ComfortObservation;
        :hasSimpleResult ?val.

Event BluePrint

Static data

Reasoning-enabled Rewriting

Query

?obs a :ComfortObservation;
        :hasSimpleResult ?val.
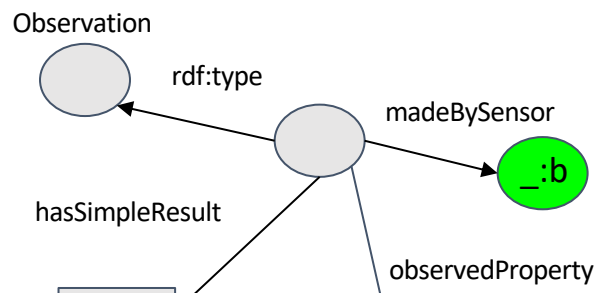
Rewritten Query

?obs a :Observation;
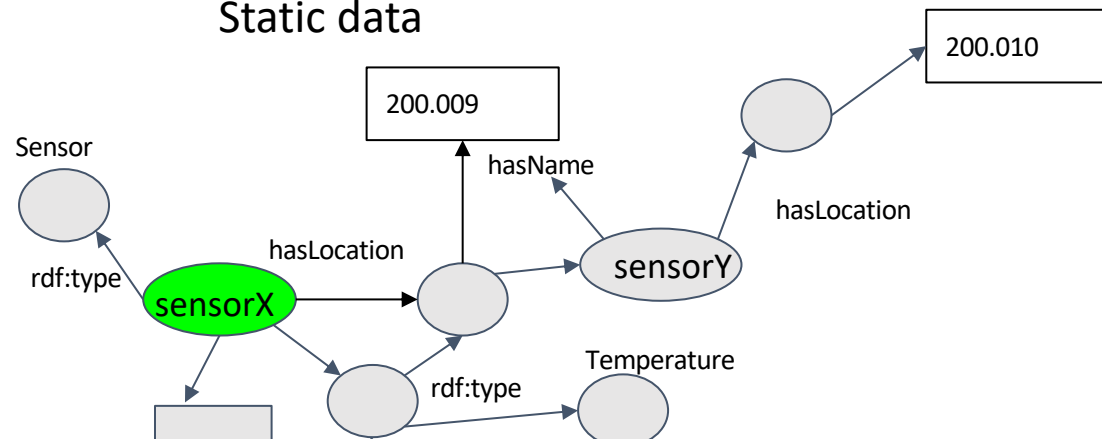   :madeBySensor :sensorX;
        :hasSimpleResult ?val.

Optimized Rules

R3'': {?x a :Observation, ?x :madeBySensor :sensorX}
    => {?x a :ComfortObservation}

Event BluePrint
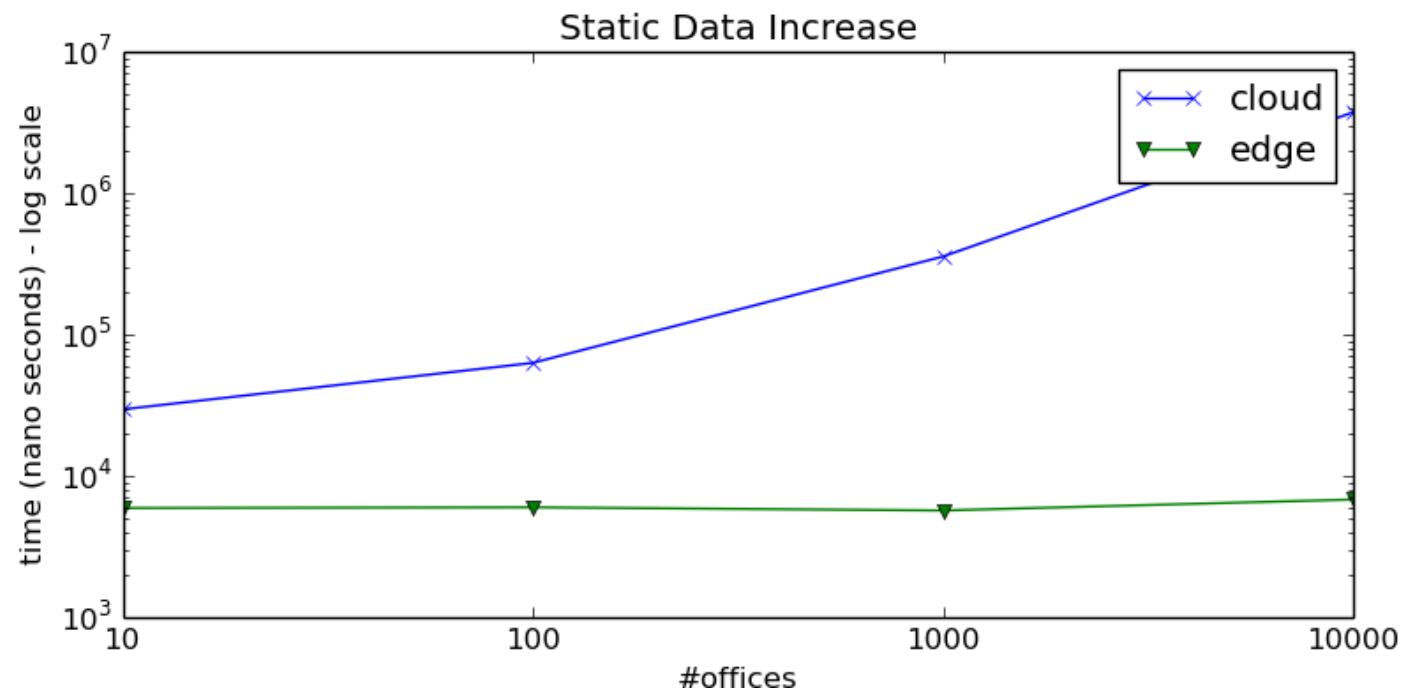

Static data

Implementation

RoXi

SHAROQ Github

RoXi Github

Come check out the poster during the poster & demo session!

Evaluation
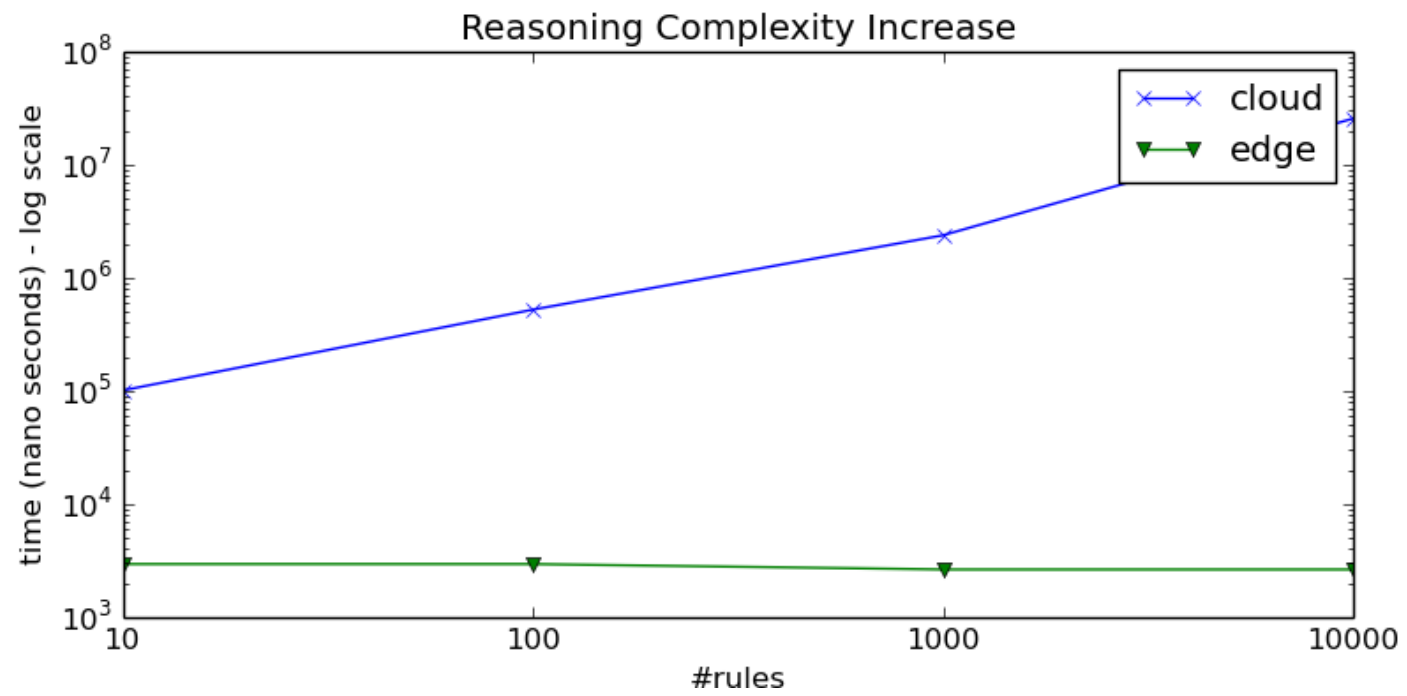

Static Data Increase

Evaluation

Evaluation

Reasoning Complexity Increase
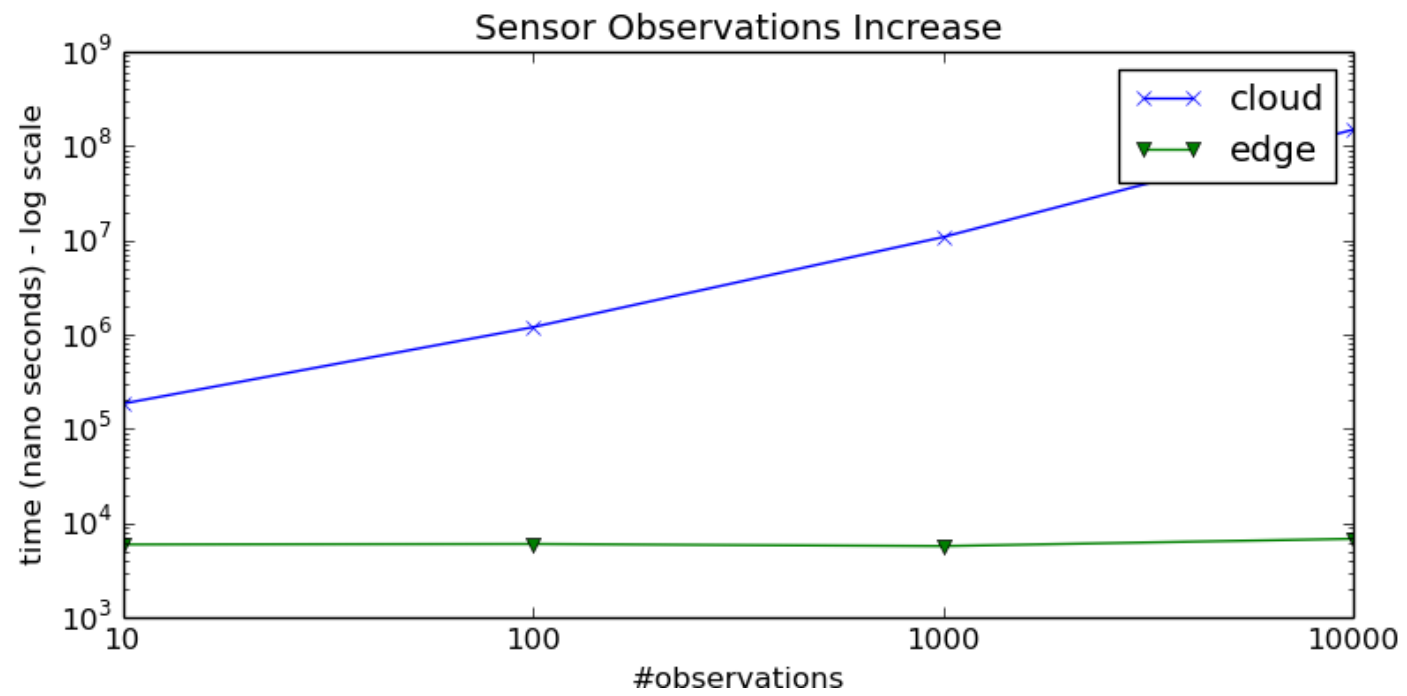
Conclusion

Top-down and bottom-up optimization to rewrite rules/queries
in order to enable efficient and privacy-aware edge analytics

Pieter.Bonte@UGent.be