POLITECNICO
MILANO 1863

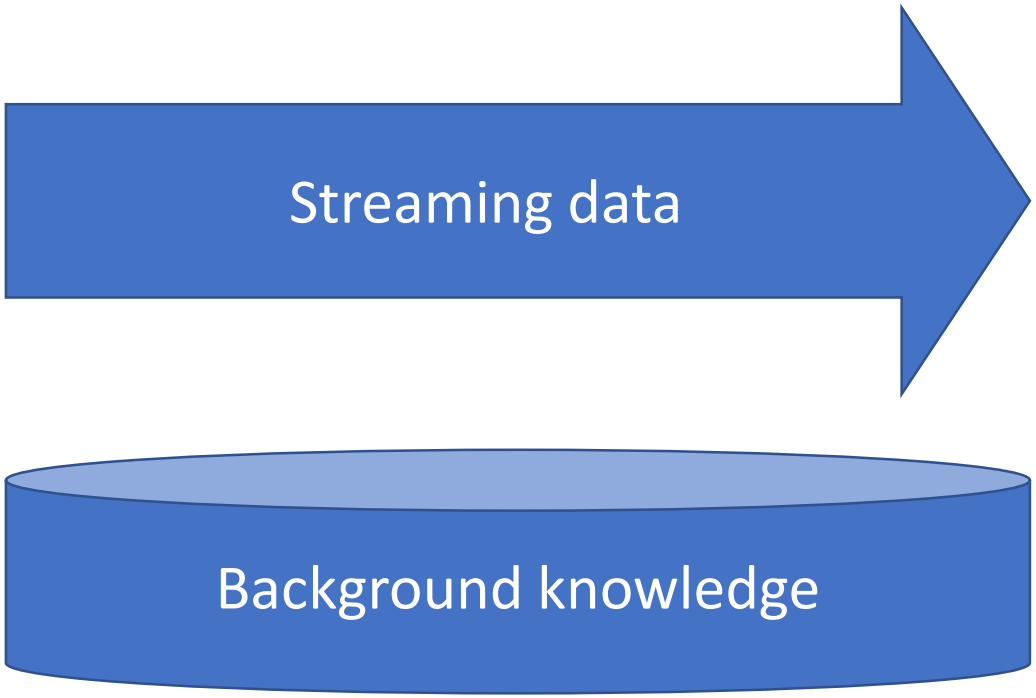
University of Zurich[UZH]

# Efficient Temporal Reasoning on Streams of Events with DOTR

Alessandro Margara, Gianpaolo Cugola

Dario Collavini, Daniele Dell'Aglio

# Stream reasoning



# Stream reasoning applications

Social Media

Smart Cities

Fraud Detection

System Monitoring

“It's a Streaming World! Reasoning upon Rapidly Changing Information”
E. Della Valle, S. Ceri, F. van Harmelen, D. Fensel, Intelligent Systems, 2009

“Streaming the Web: Reasoning over Dynamic Data”
A. Margara, J. Urbani, F. van Harmelen, H. Bal, J. Web Semantics, 2014

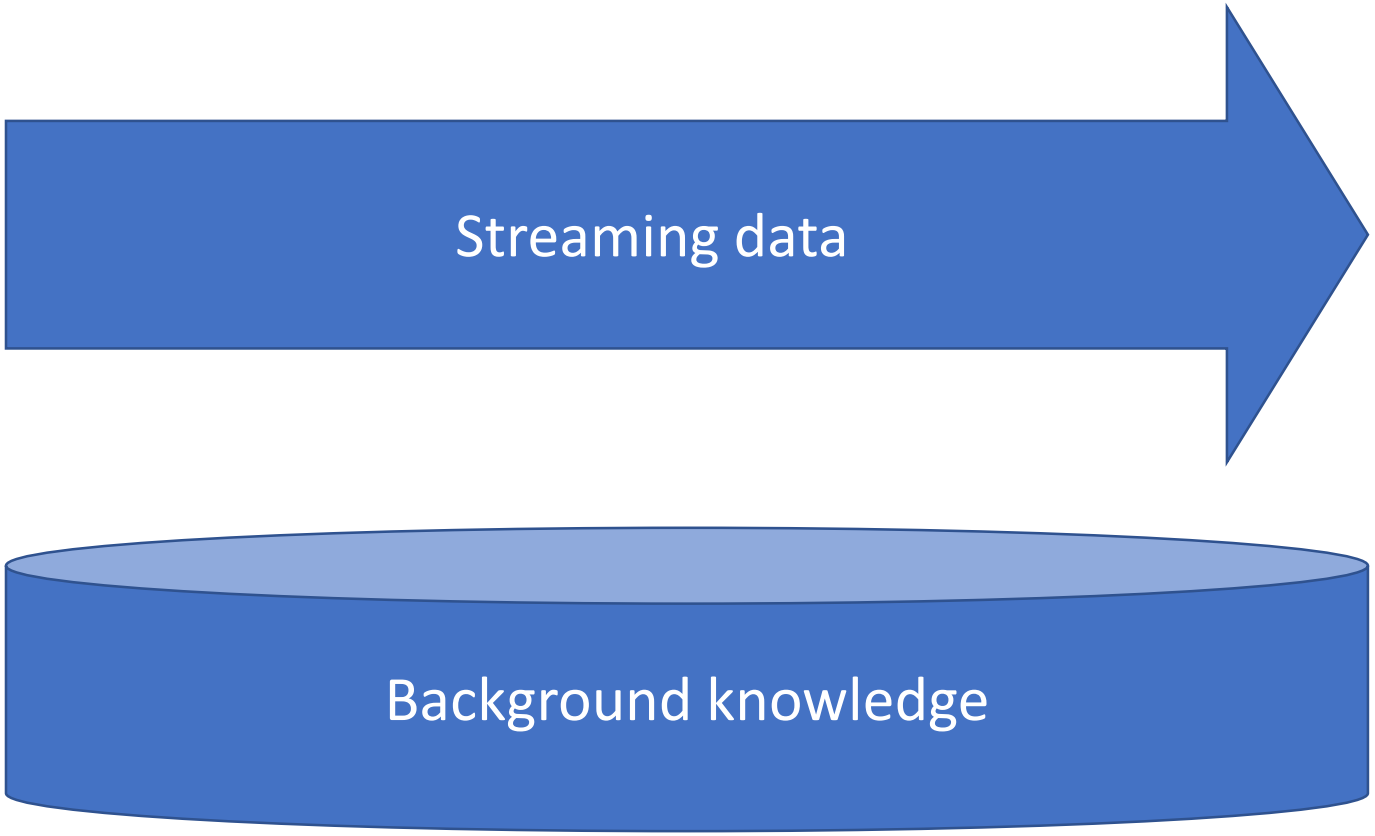
# Stream reasoning requirements

1. Flexible data model to integrate data from heterogeneous sources
2. Support for background knowledge that describes the application domain
3. High throughput and low latency
4. Expressive (*temporal*) reasoning

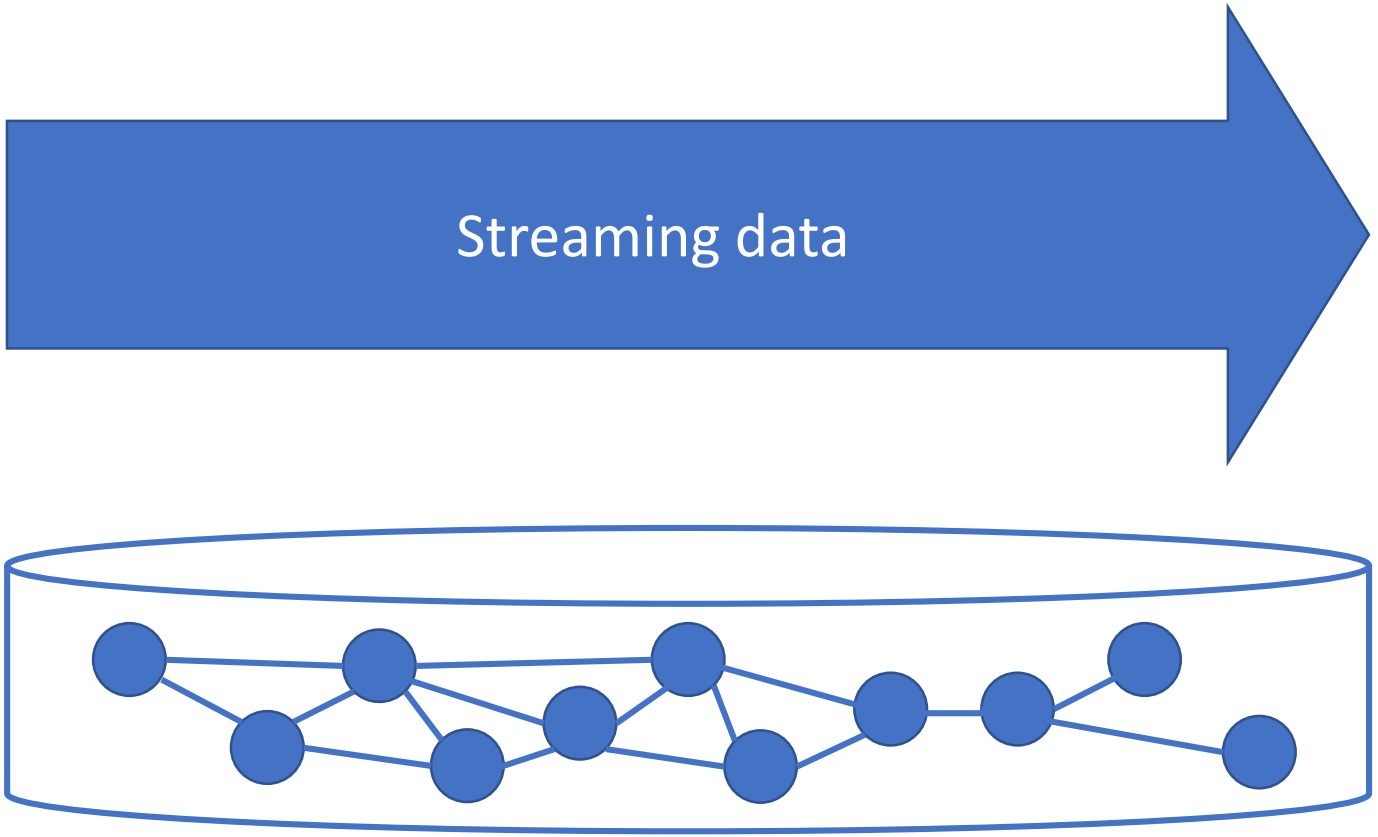
# State of the art

| | Stream Processing | RDF Stream Processing / Stream Reasoning |
|---|---|---|
| 1. Flexible data model | Ad-hoc, structured | RDF ✓ |
| 2. Background knowledge | Not always<br>No reasoning | Yes ✓ |
| 3. Performance | High throughput<br>Low latency ✓ | Variable |
| 4. Expressive reasoning | Only temporal reasoning (CEP) | Limited temporal reasoning |

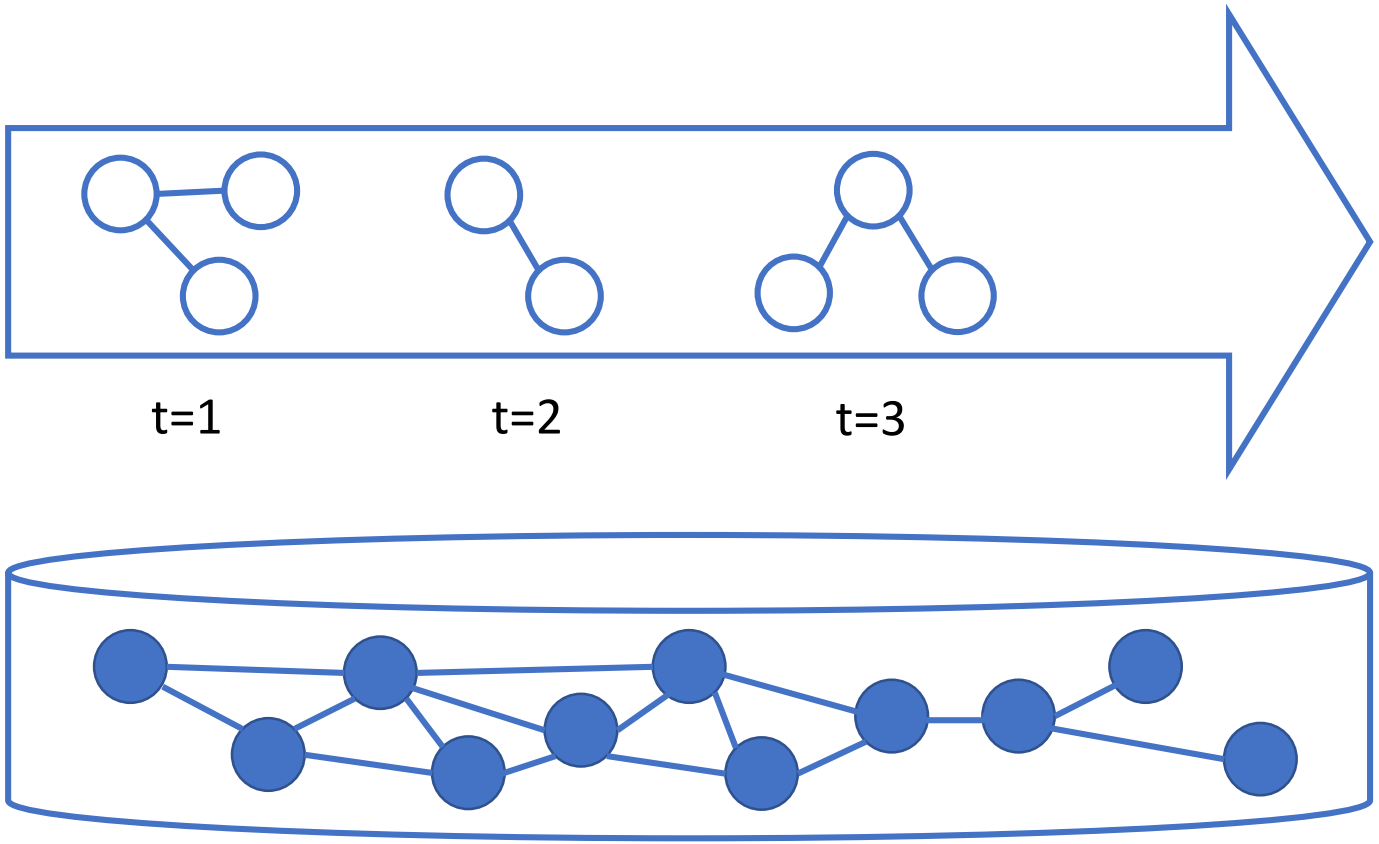
# Stream reasoning model



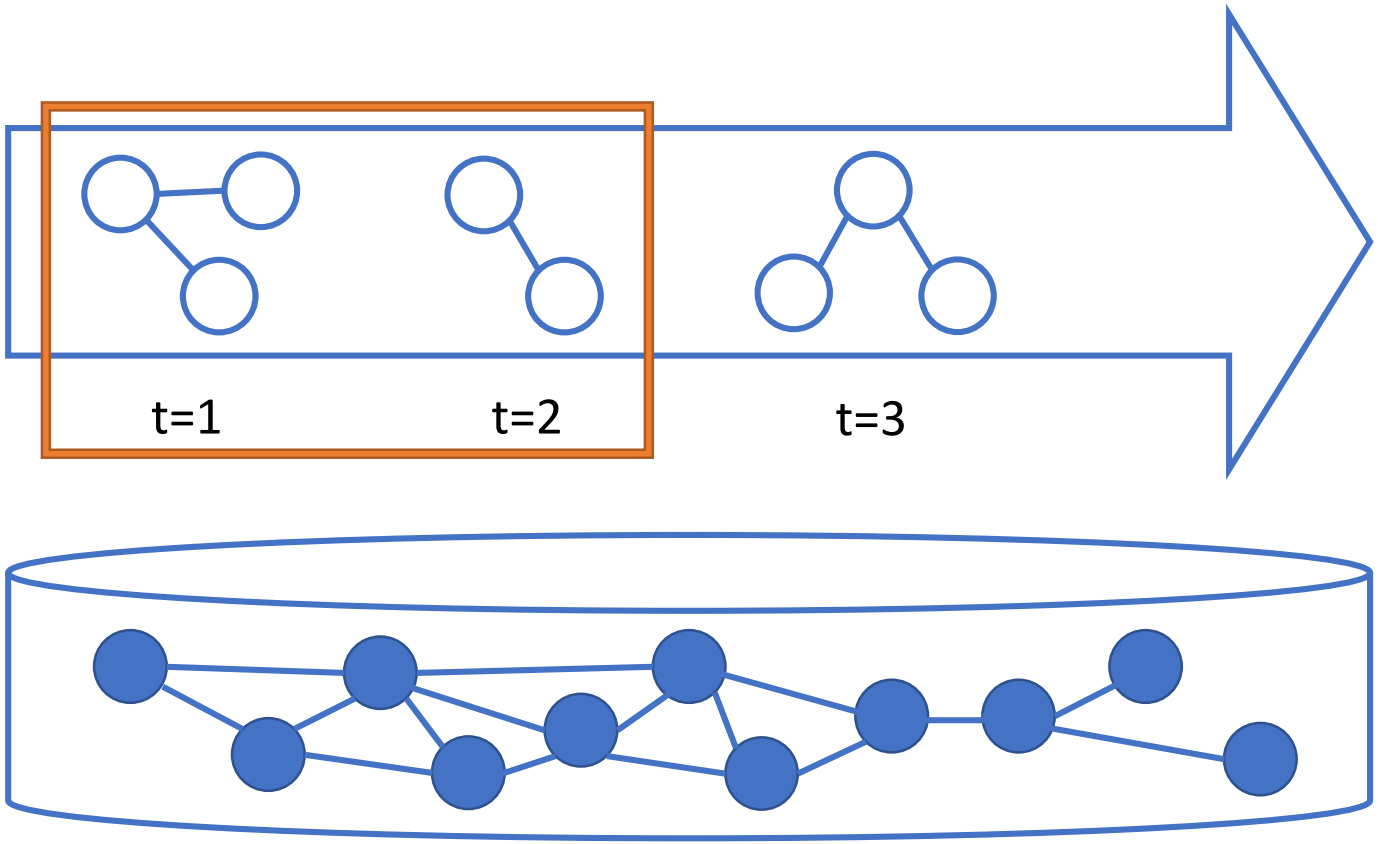
# Stream reasoning model



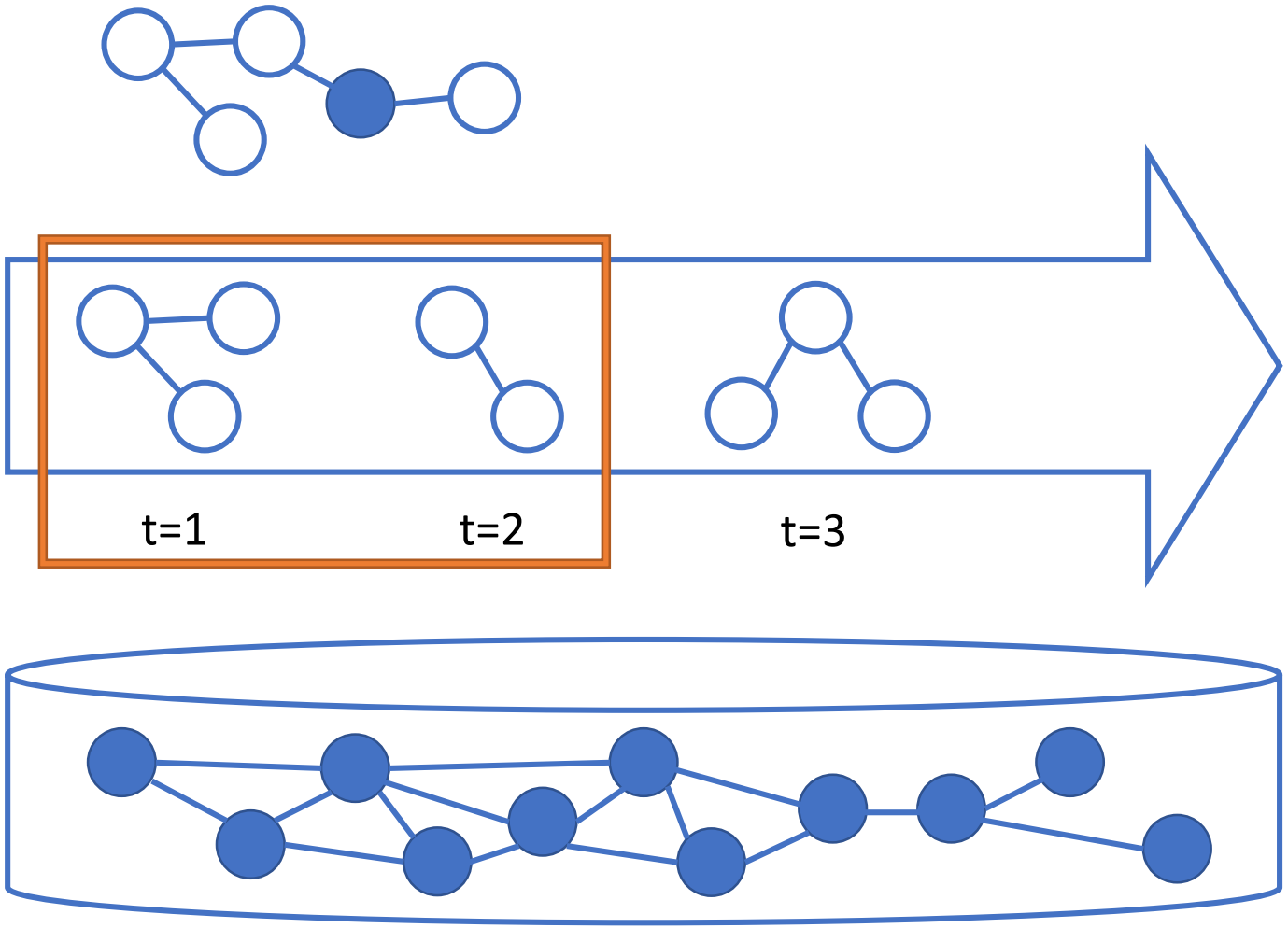
# Stream reasoning model



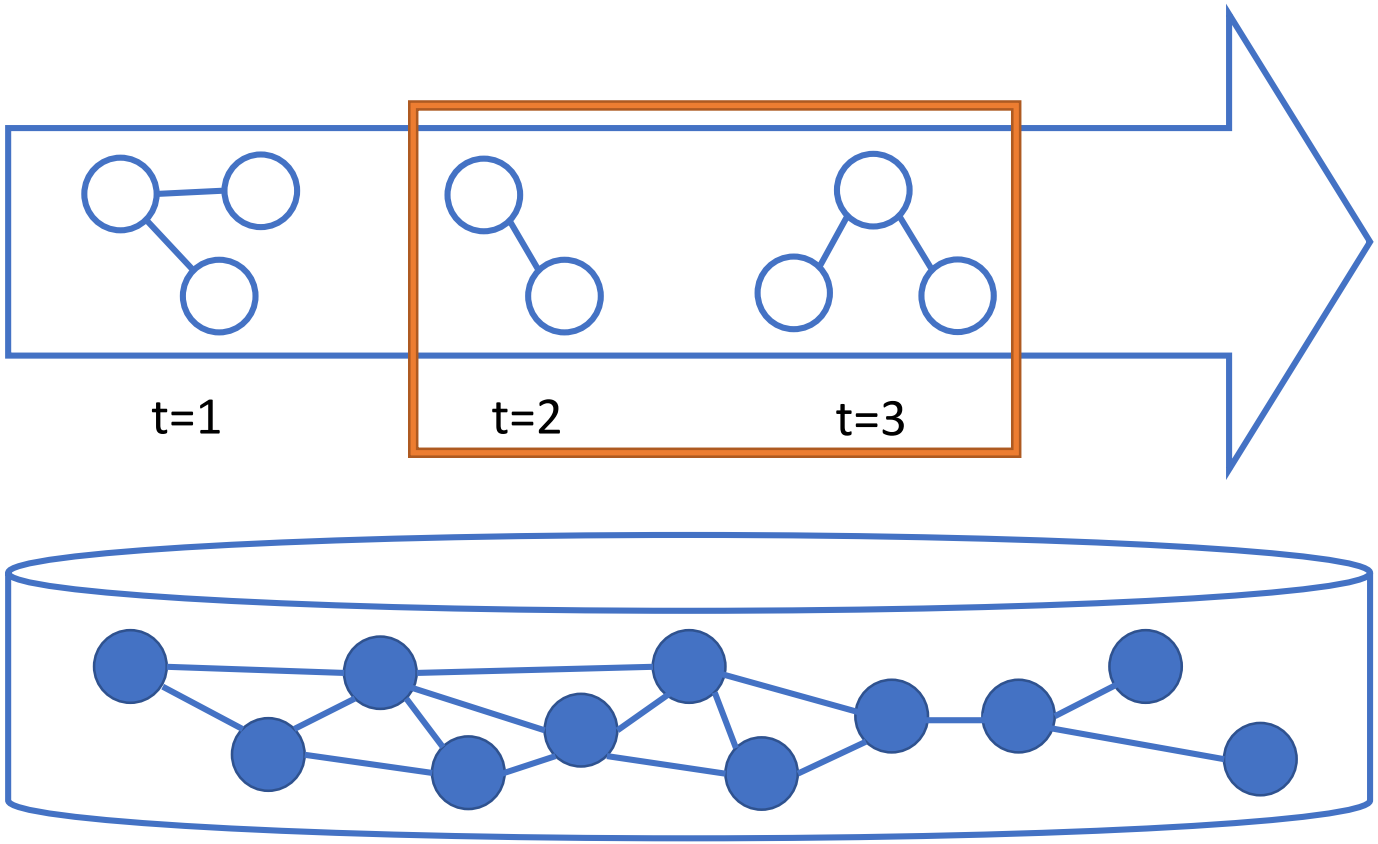
# Stream reasoning model



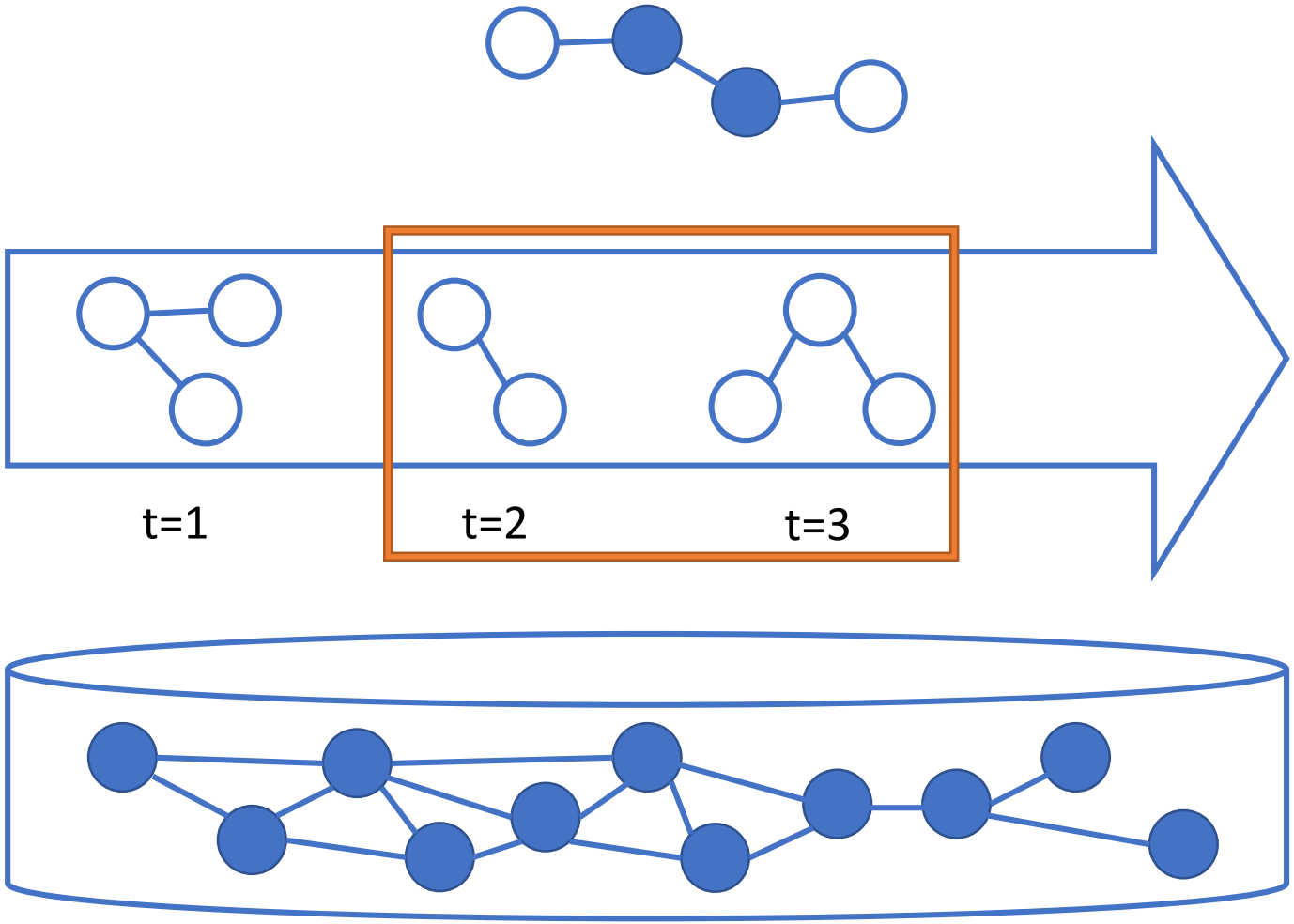
# Stream reasoning model



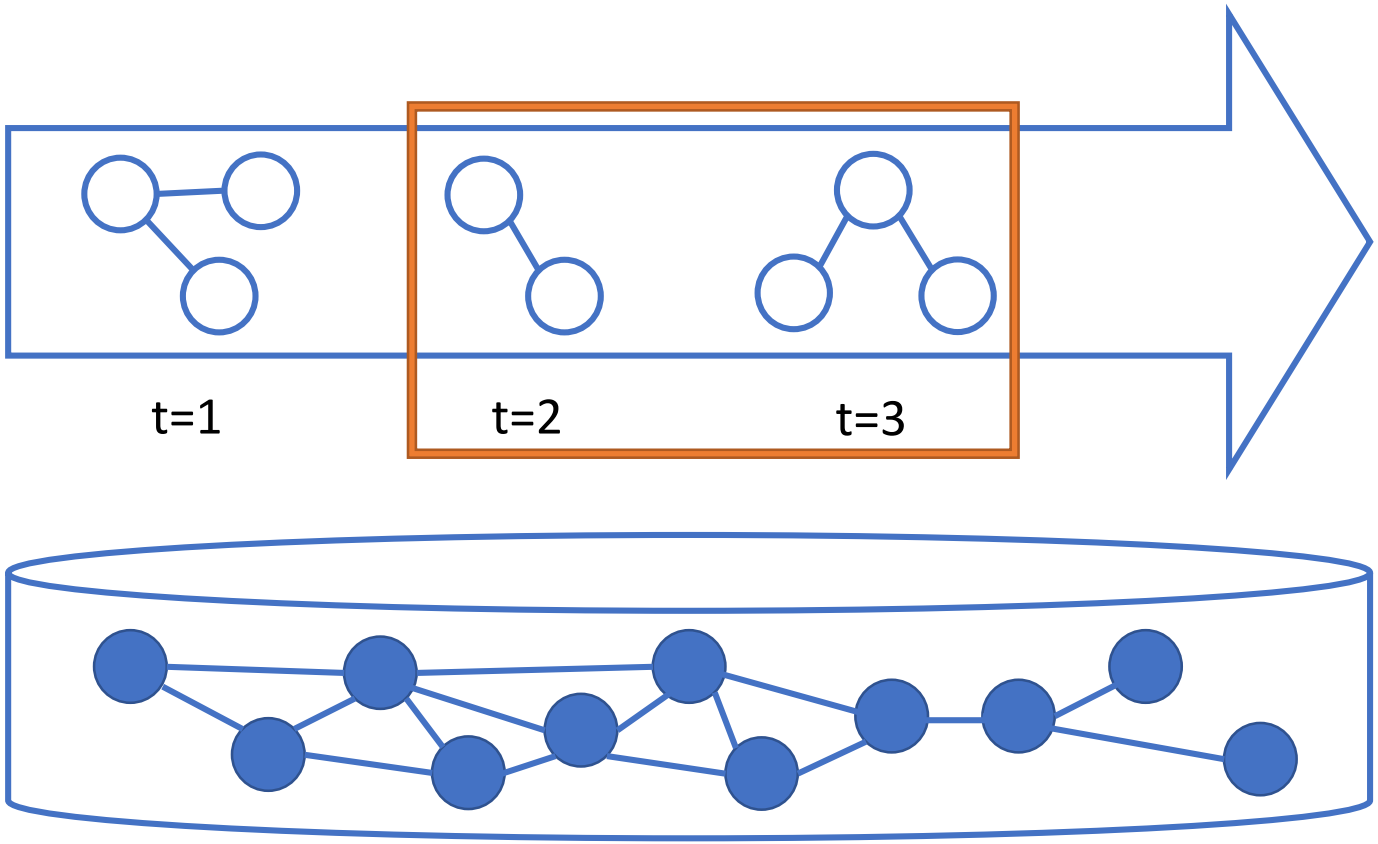
# Stream reasoning model



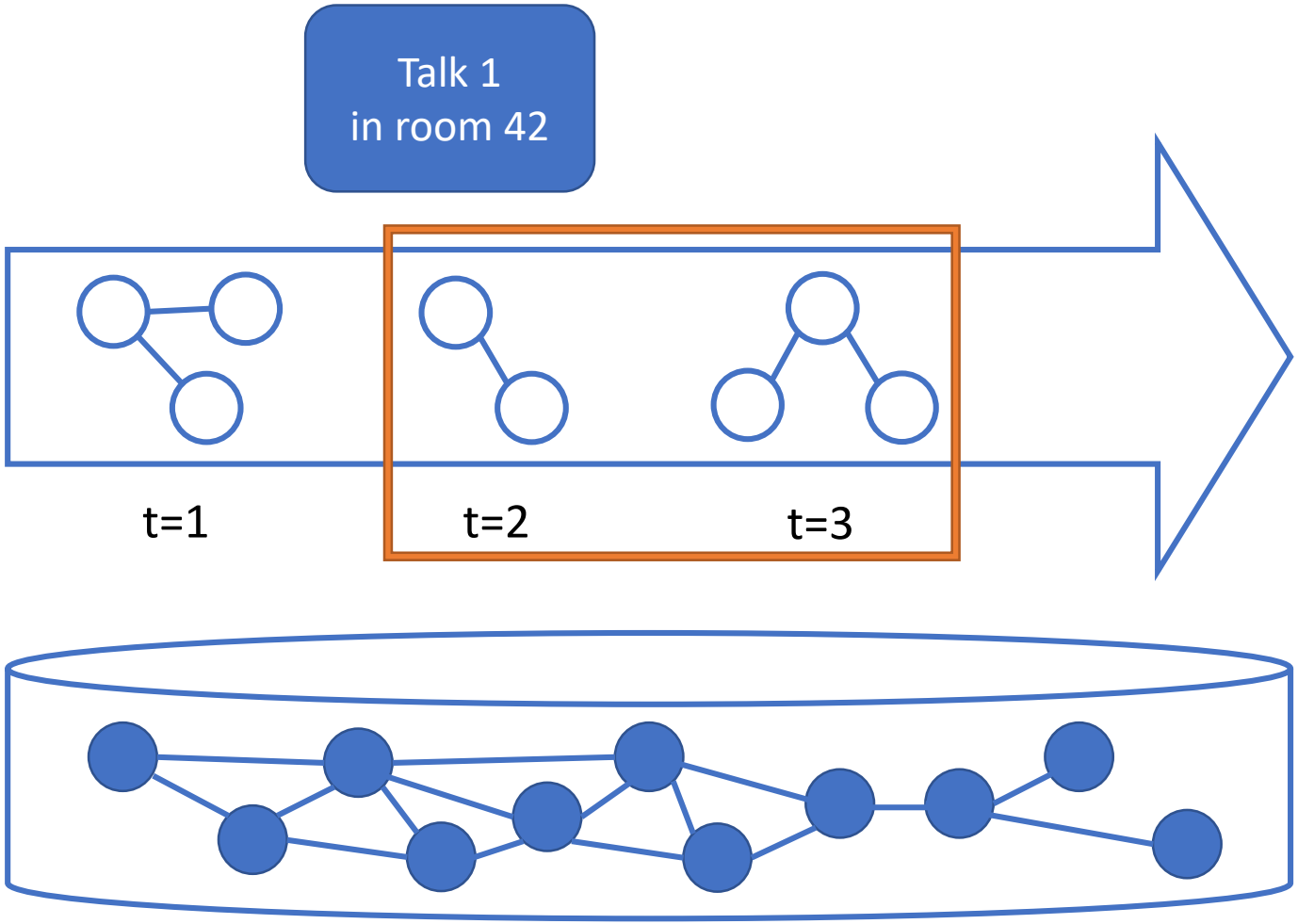
# Stream reasoning model



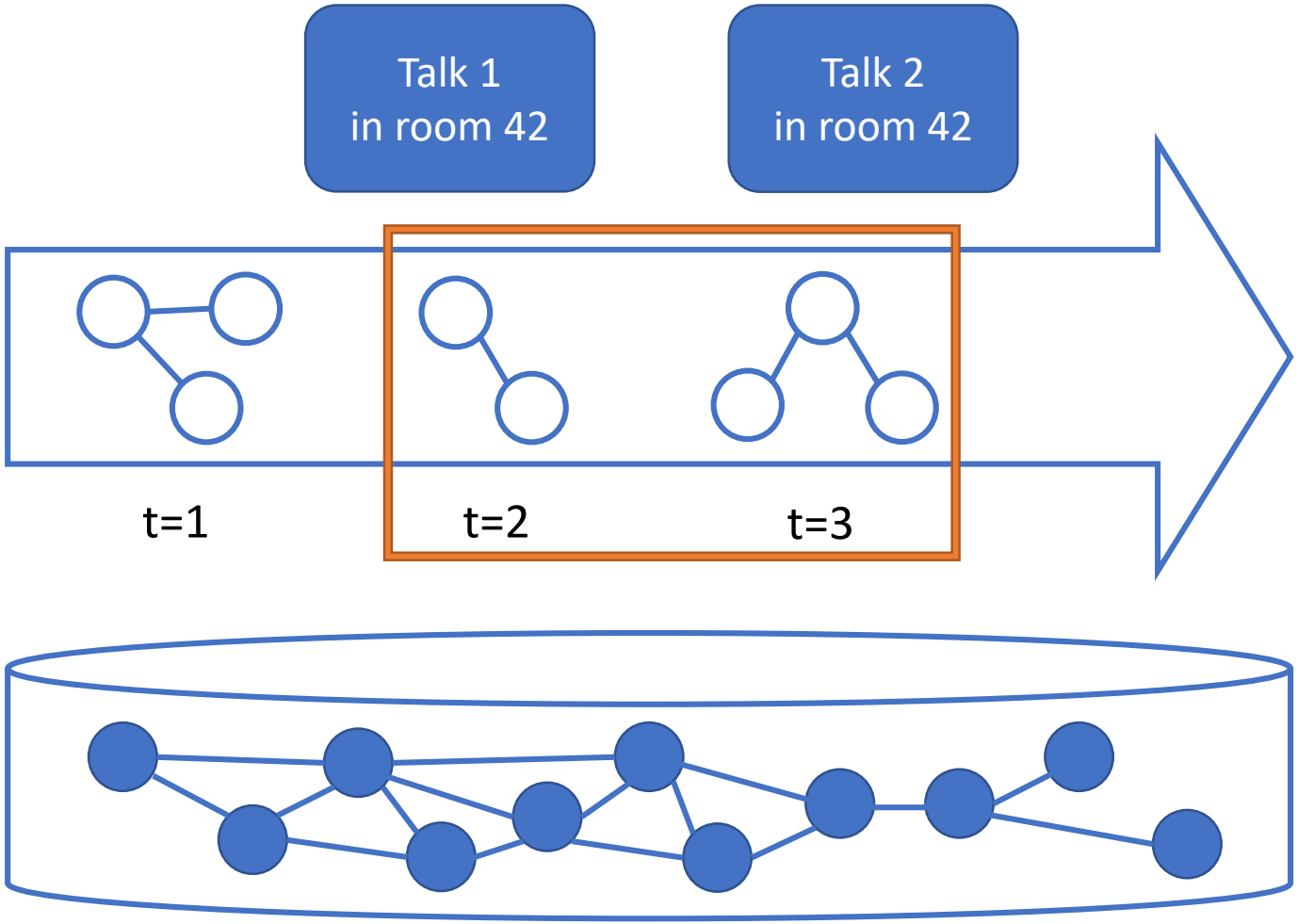
# Temporal reasoning problems



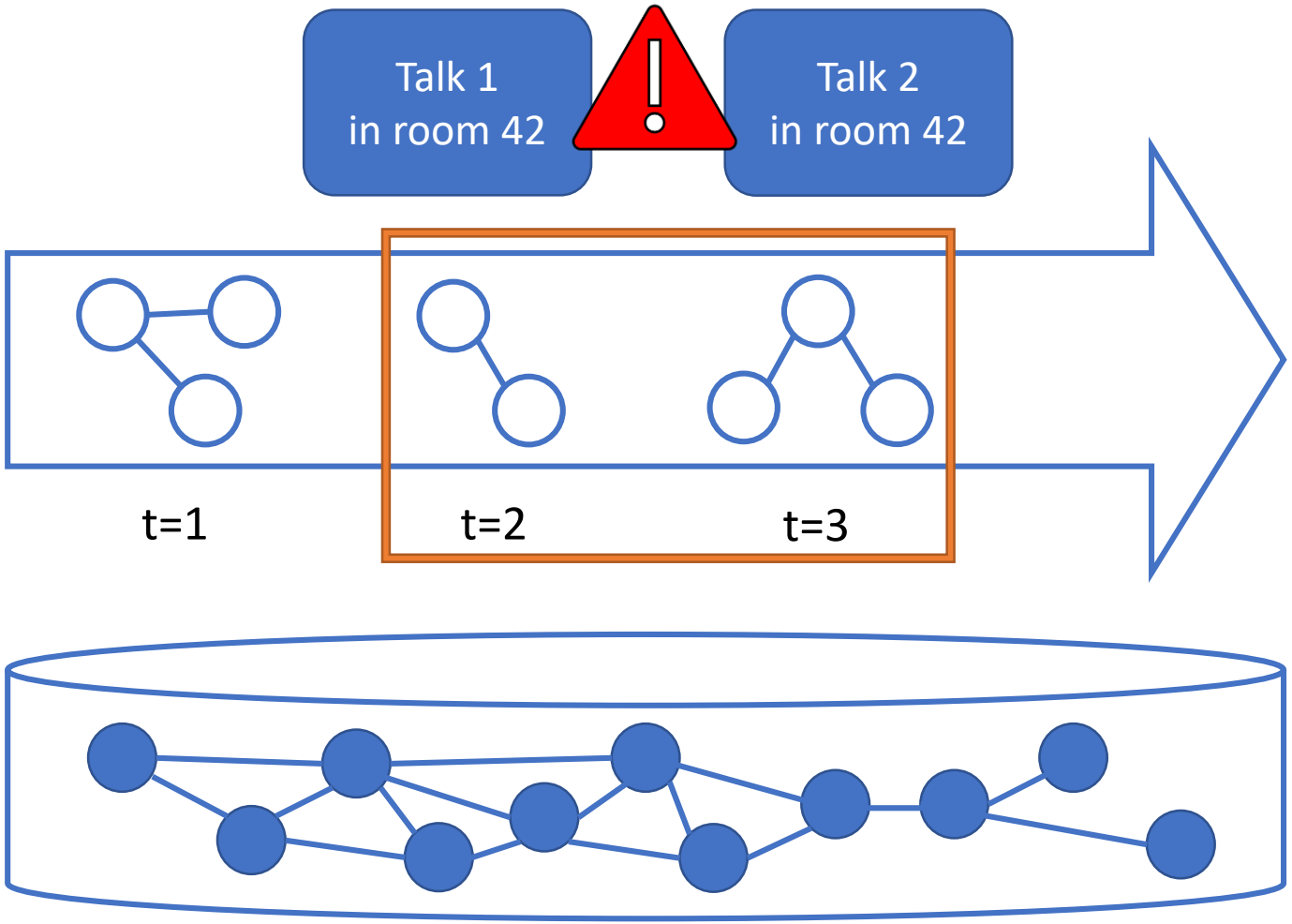
# Temporal reasoning problems



# Temporal reasoning problems

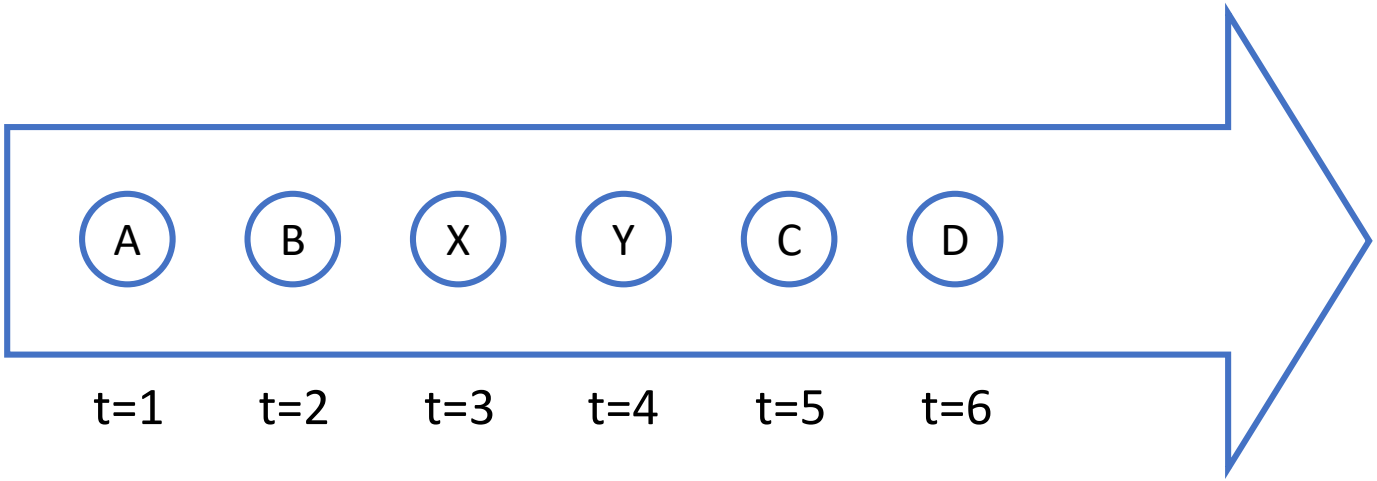


# Temporal reasoning problems



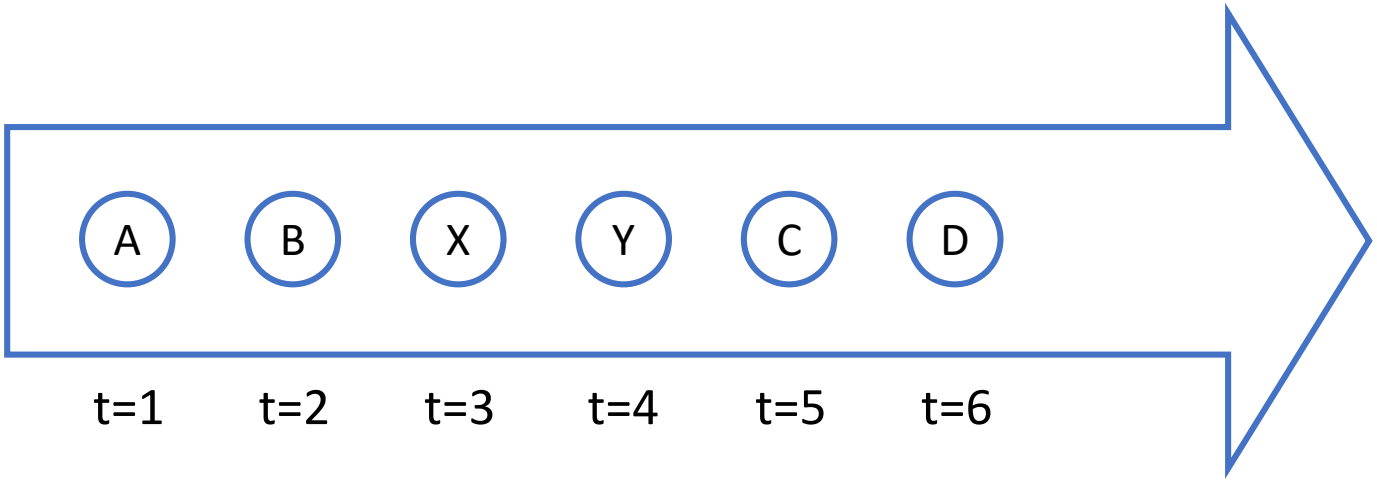
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units



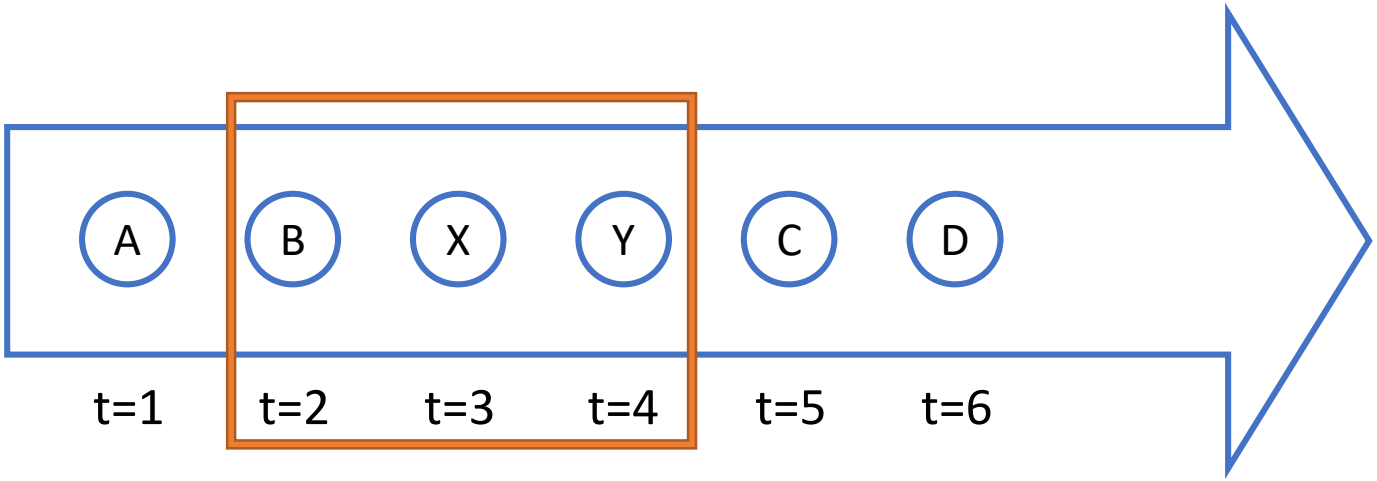
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 1



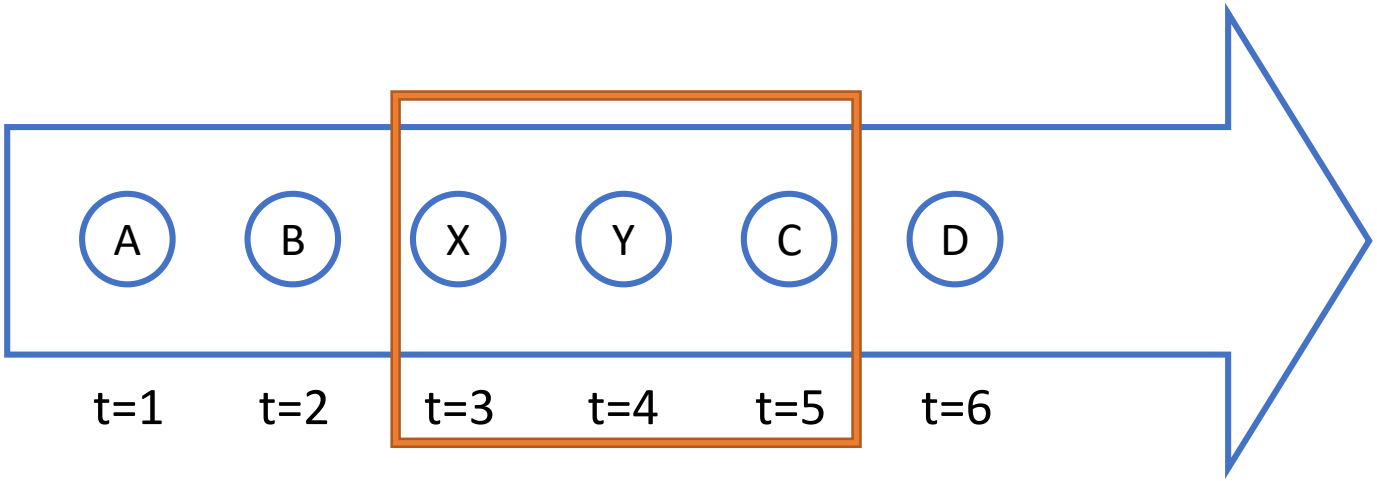
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 1



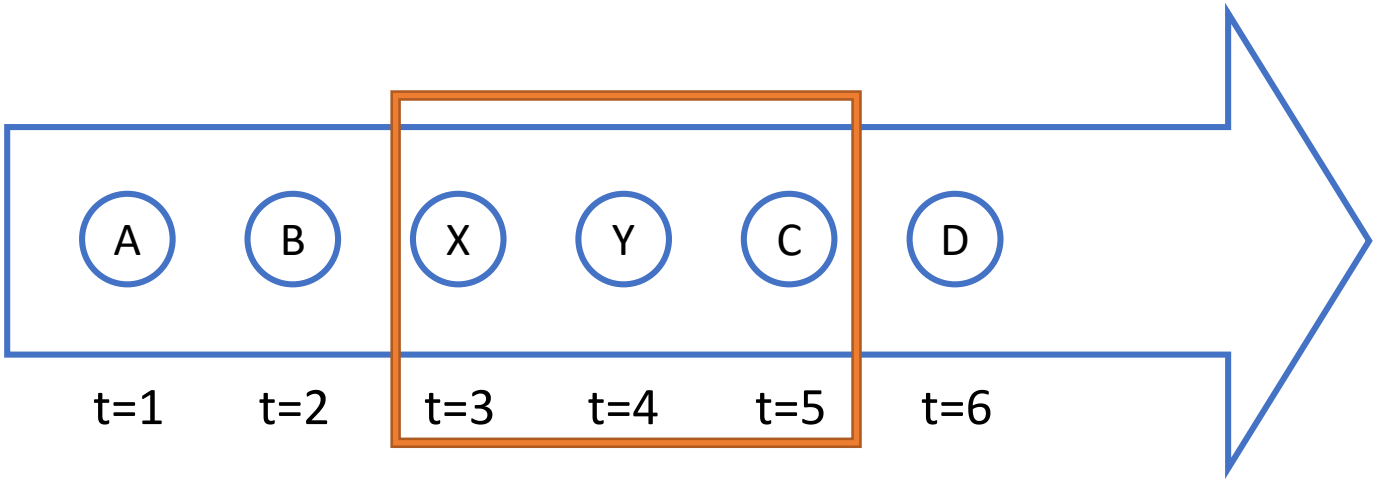
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 1



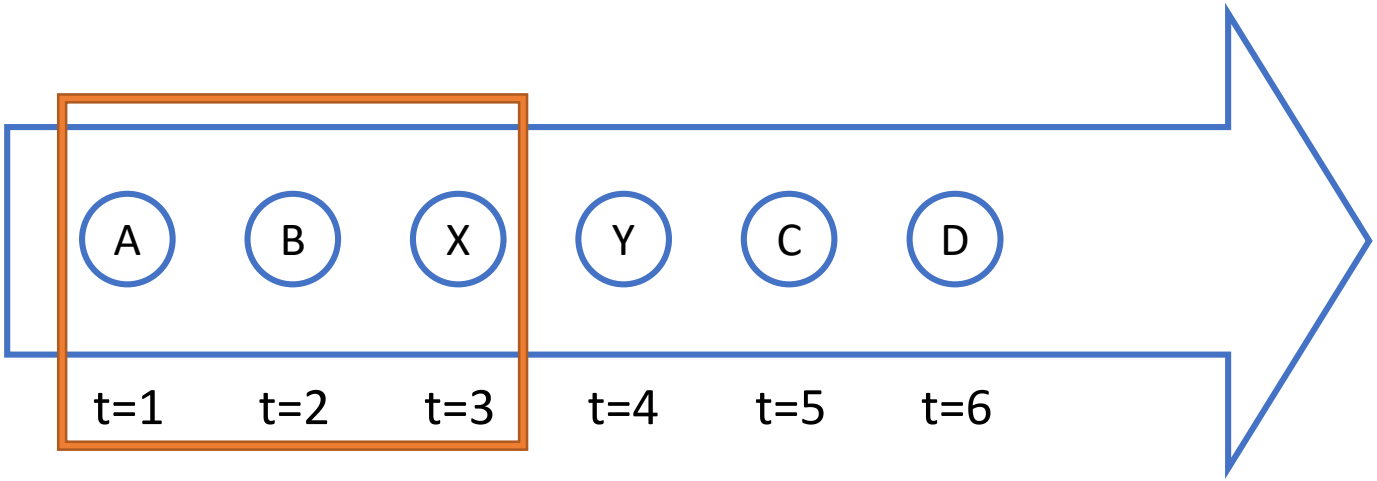
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 1
- Same occurrence detected twice!



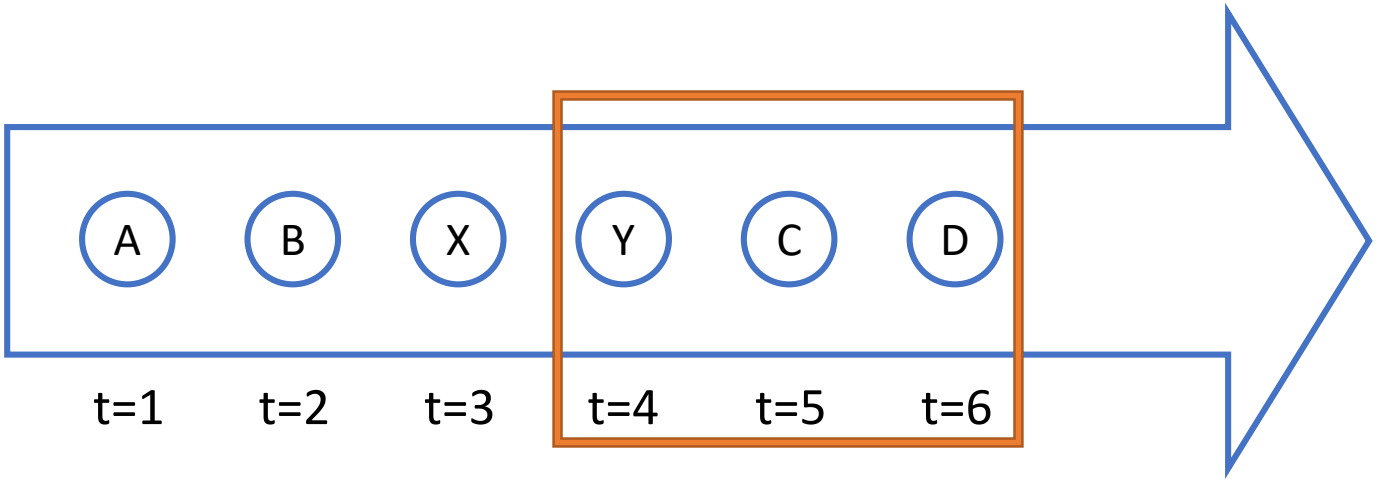
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 3



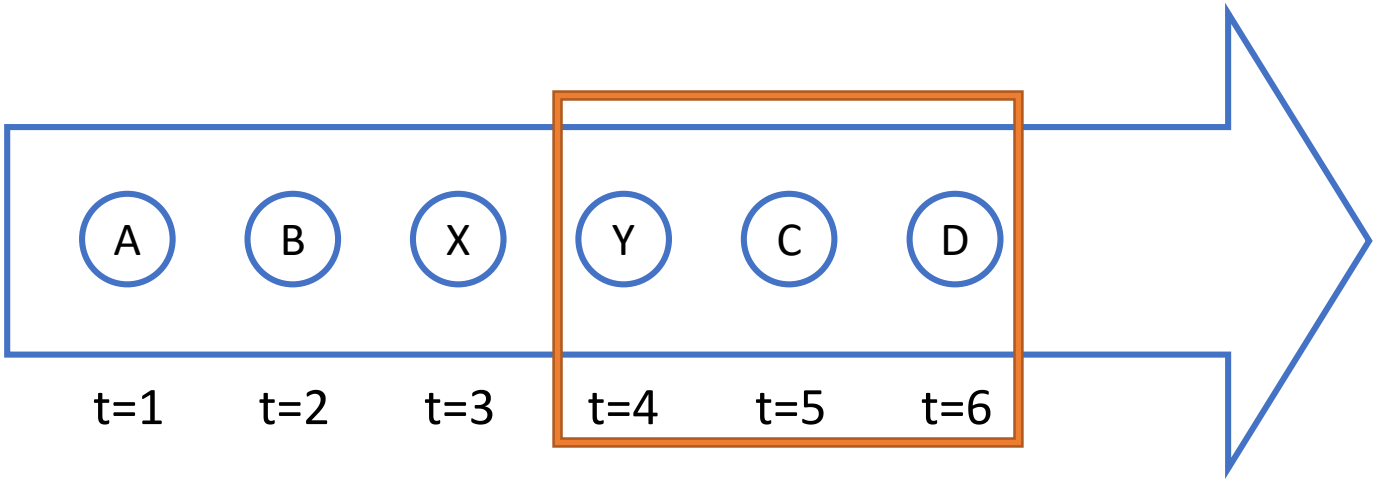
# Temporal reasoning problems

- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 3



# Temporal reasoning problems

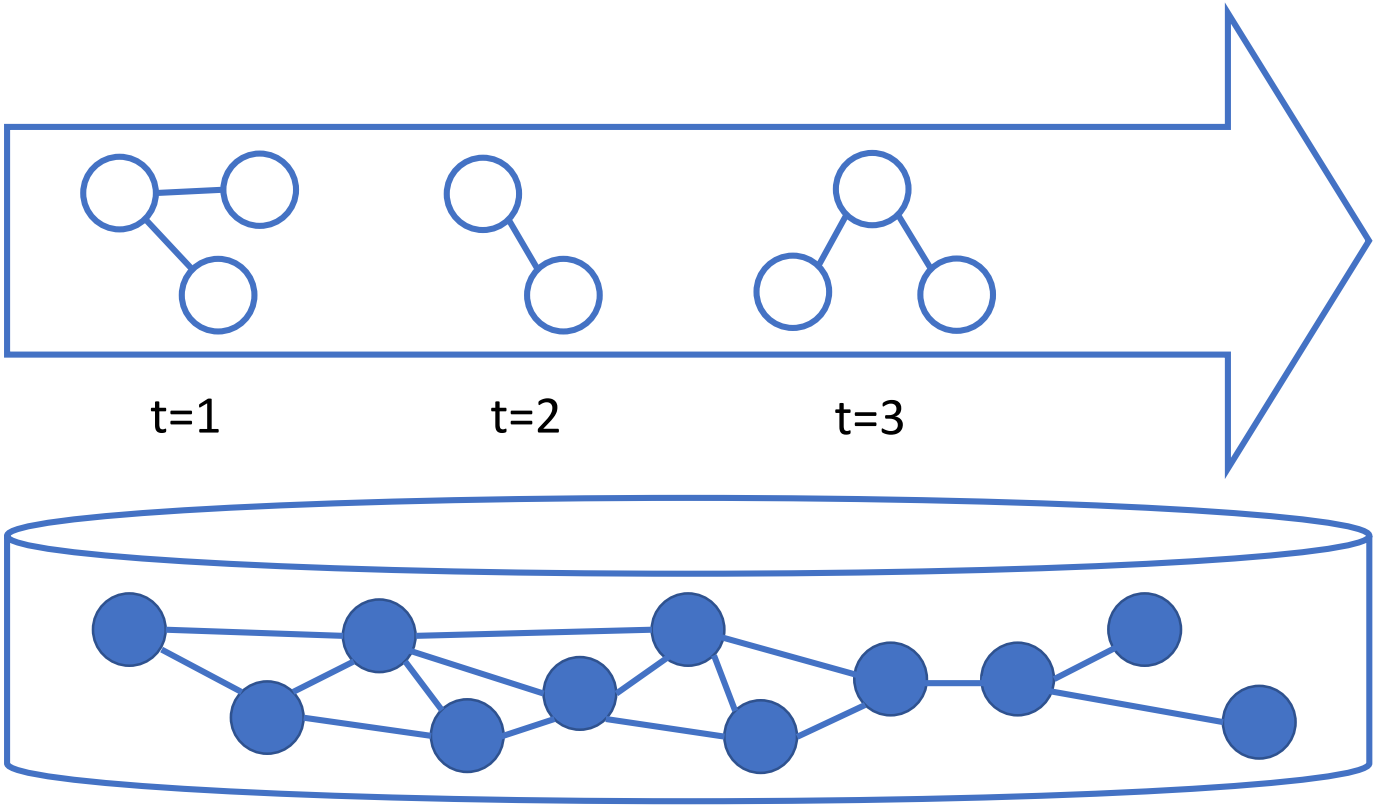
- We want to detect X followed by Y within 3 time units
- Window of size 3 and slide 3
- Occurrence not detected!



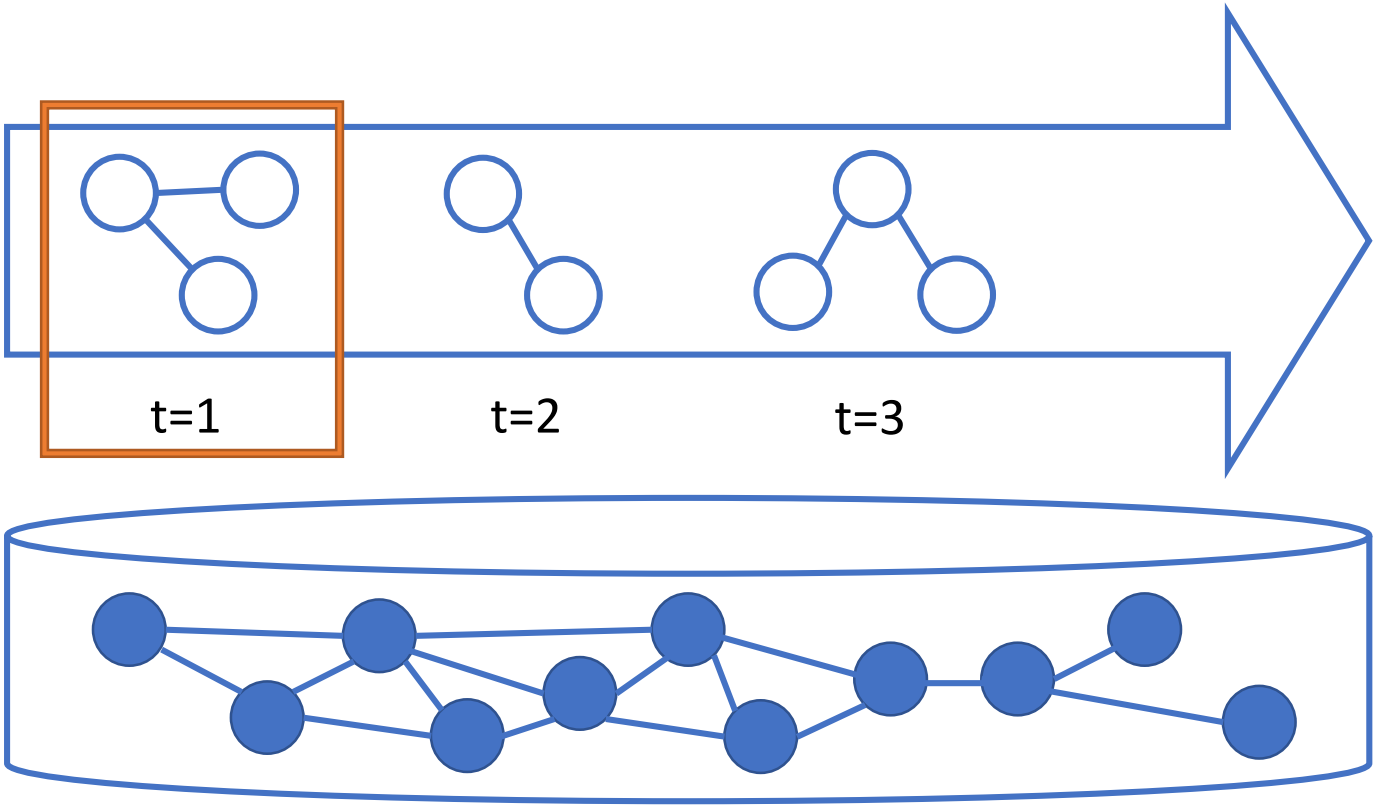
# The DOTR model

| | Stream Processing | RDF Stream Processing / Stream Reasoning |
|---|---|---|
| 1. Flexible data model | Ad-hoc, structured | RDF ✓ |
| 2. Background knowledge | Not always<br>No reasoning | Yes ✓ |
| 3. Performance | High throughput<br>Low latency ✓ | Variable |
| 4. Expressive reasoning | Only temporal reasoning (CEP) ✓ | Limited temporal reasoning |

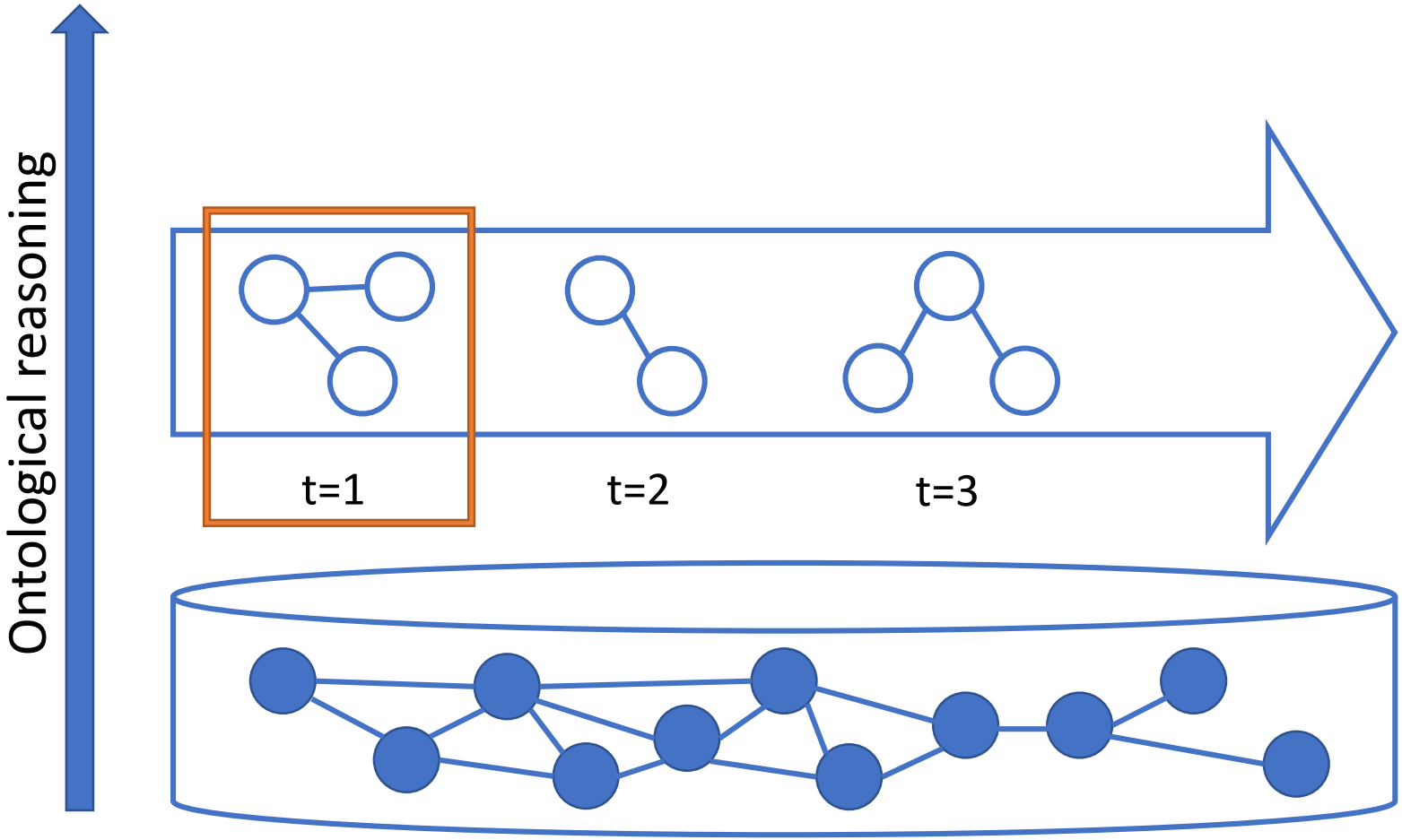
# The DOTR model



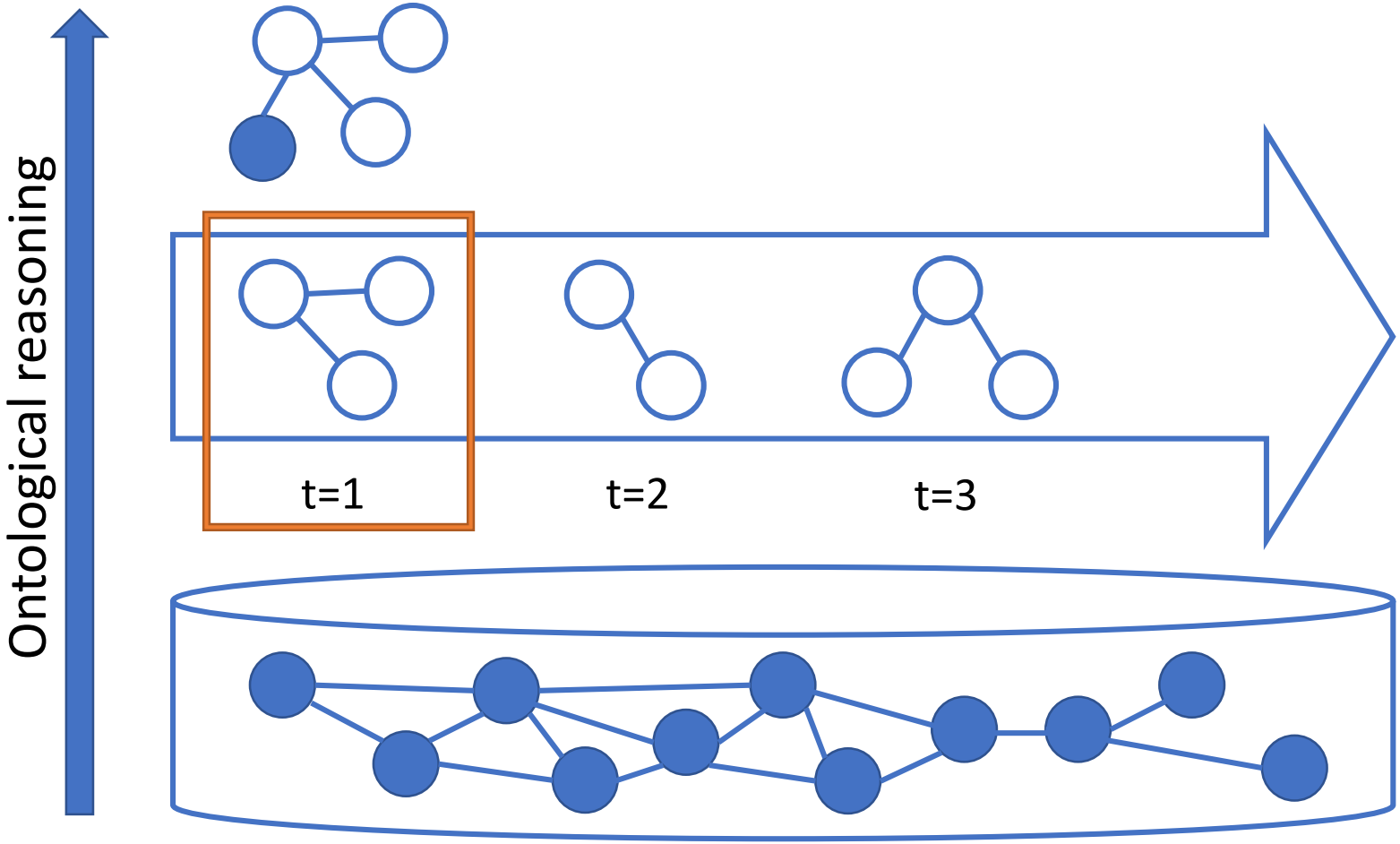
# The DOTR model



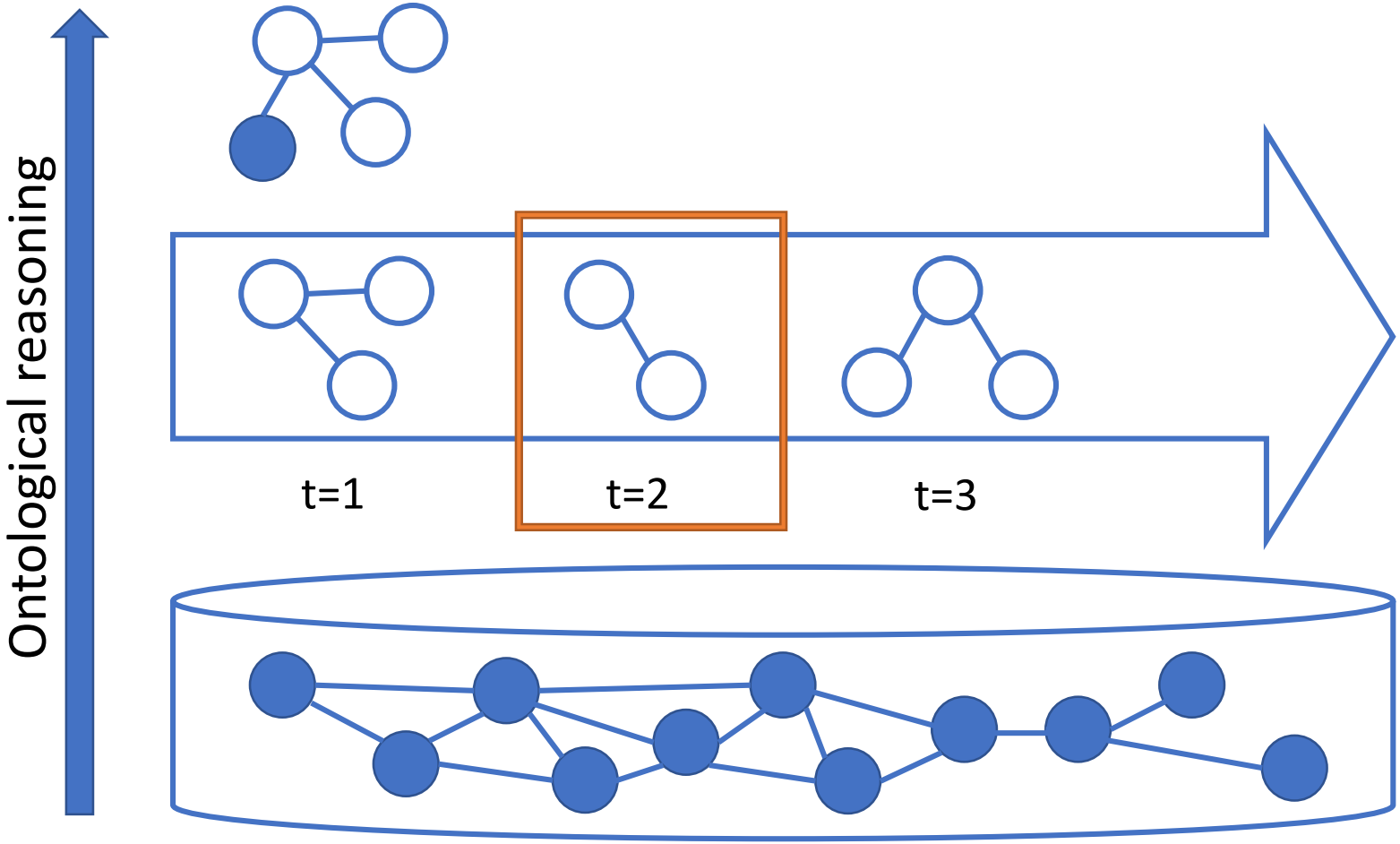
# The DOTR model



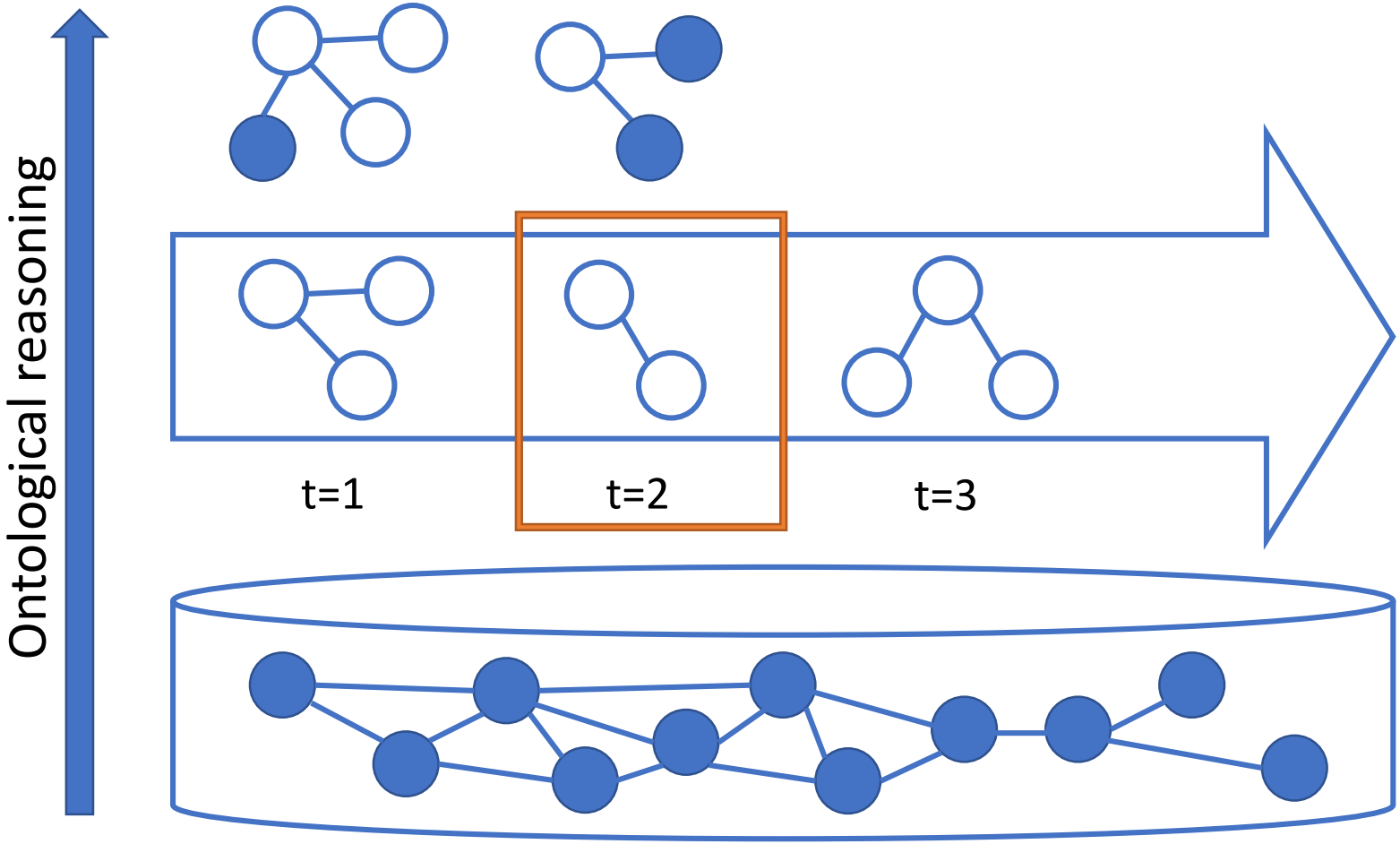
# The DOTR model



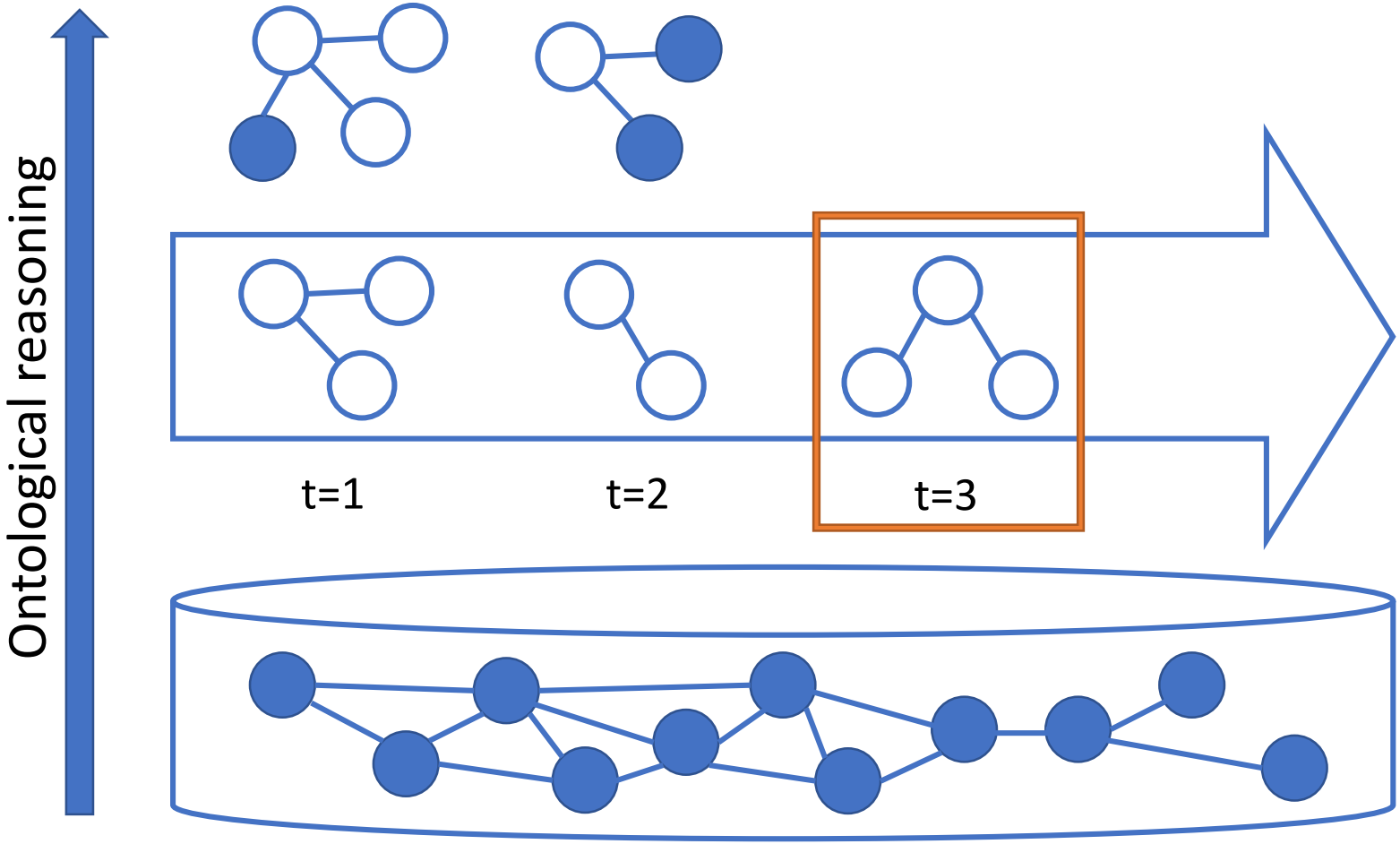
# The DOTR model



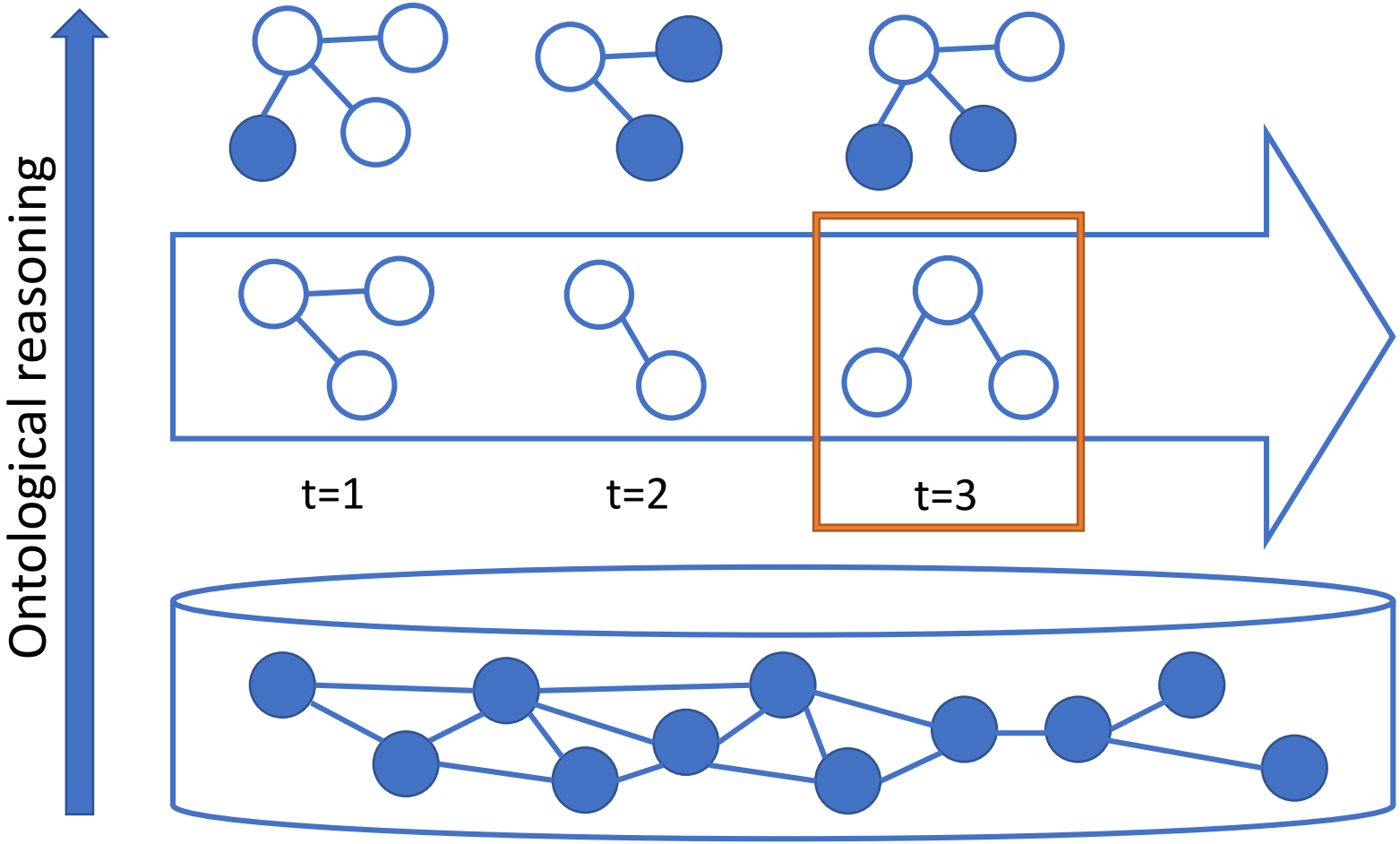
# The DOTR model



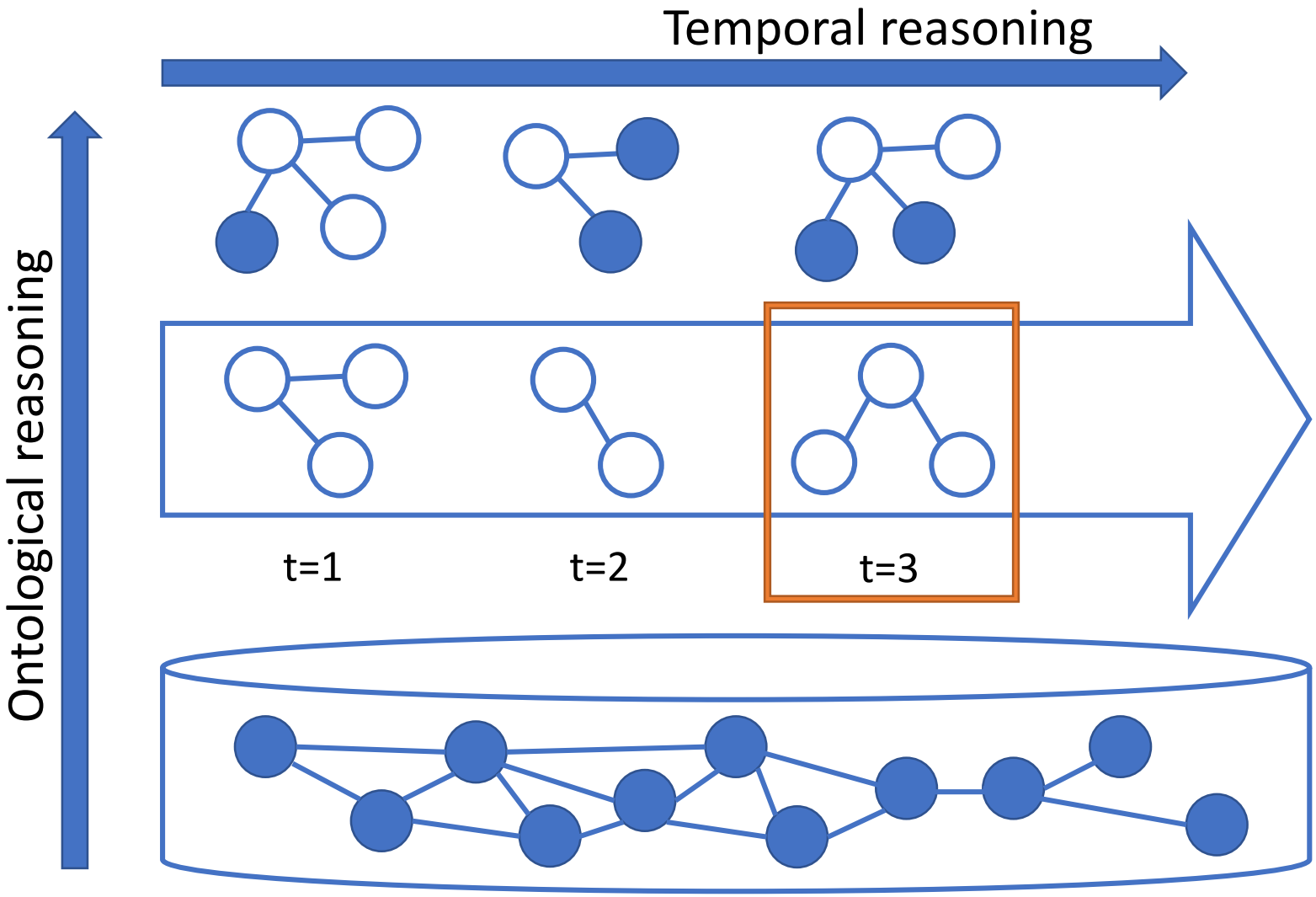
# The DOTR model



# The DOTR model

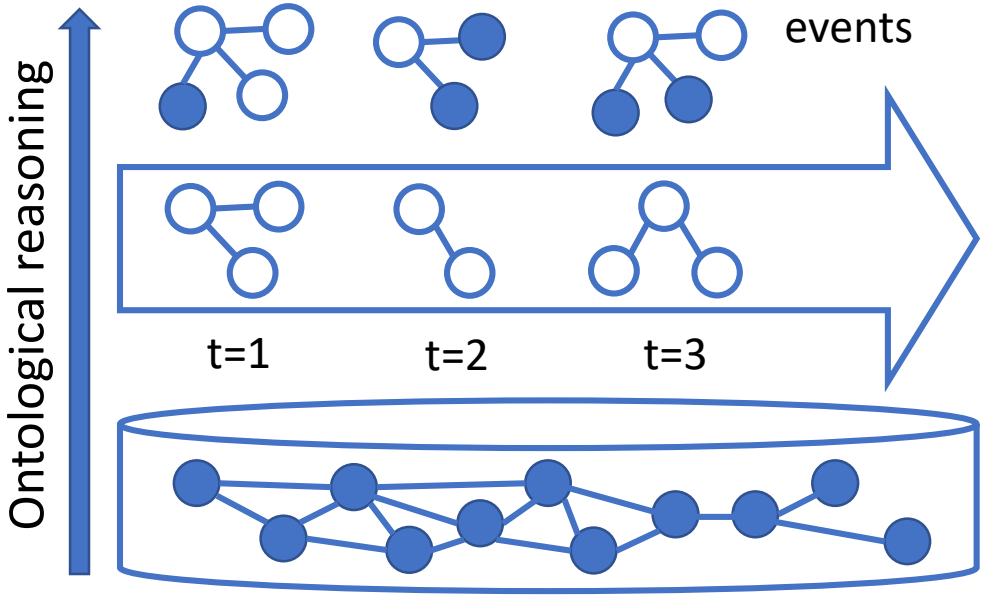


# The DOTR model



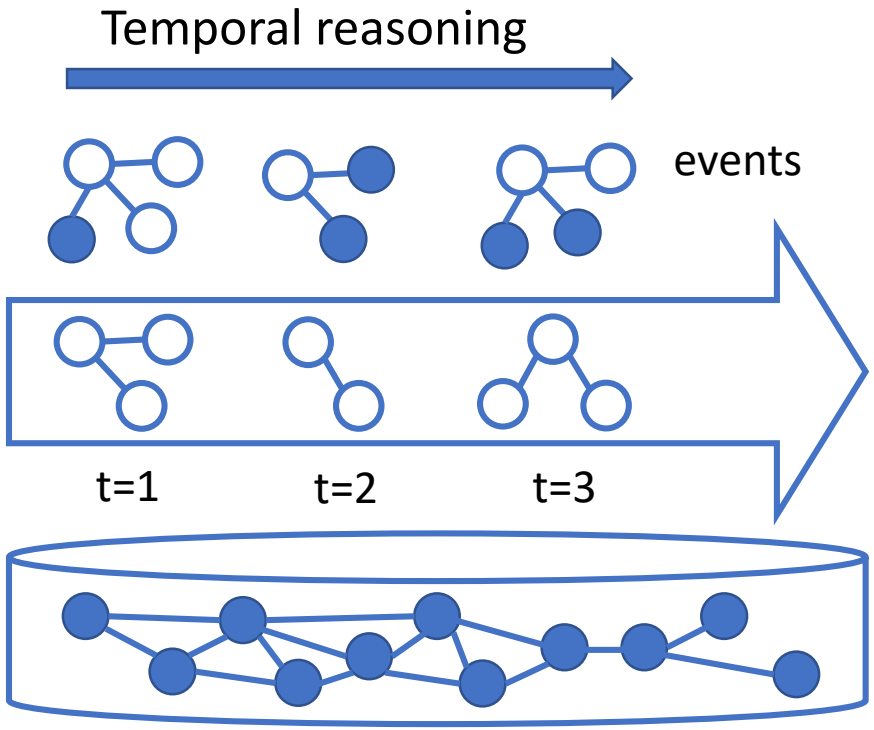
# Ontological reasoning

- Datalog rules
  - Background knowledge
    + data at time t
- SPARQL queries
  - Extract relevant facts
  - *Events*

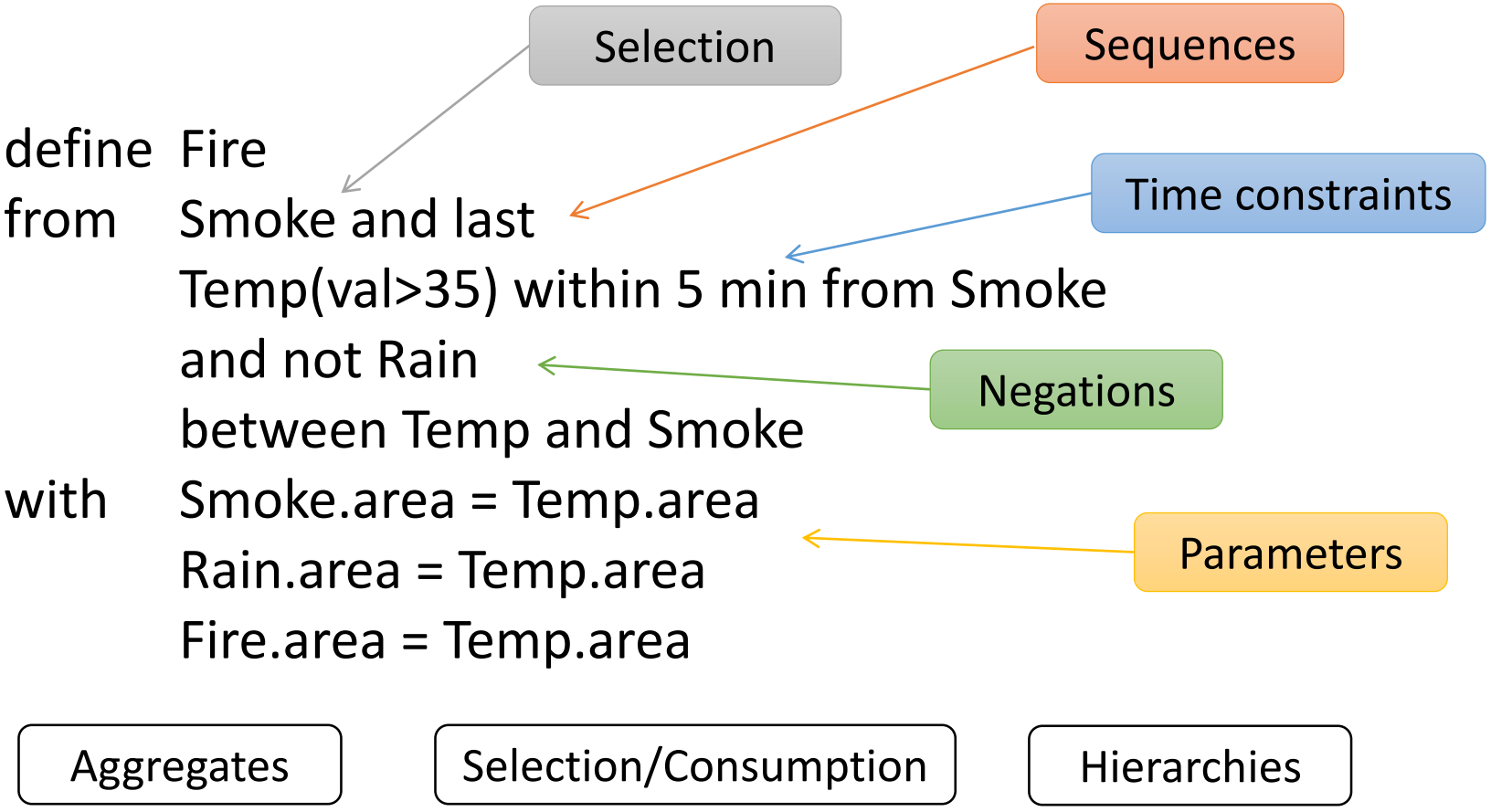


# Temporal reasoning

- Combines events that occur at different points in time …
- … into *patterns*



# Temporal reasoning (only!): TESLA




Aggregates

Selection/Consumption

Hierarchies

Temporal reasoning is formally defined through metric temporal logic formulas

# The DOTR language

**define** Fire = [dotr_id1 rdf:type :Fire. dotr_id1 :at_room ?room ]
**from** Smoke = [

```
SELECT ?read1 ?room1
WHERE {
        ?read1 rdf:type :Smoke. ?read1 :is_from_sensor ?sens1.
        ?sens1 :is_in_room ?room1 }
```

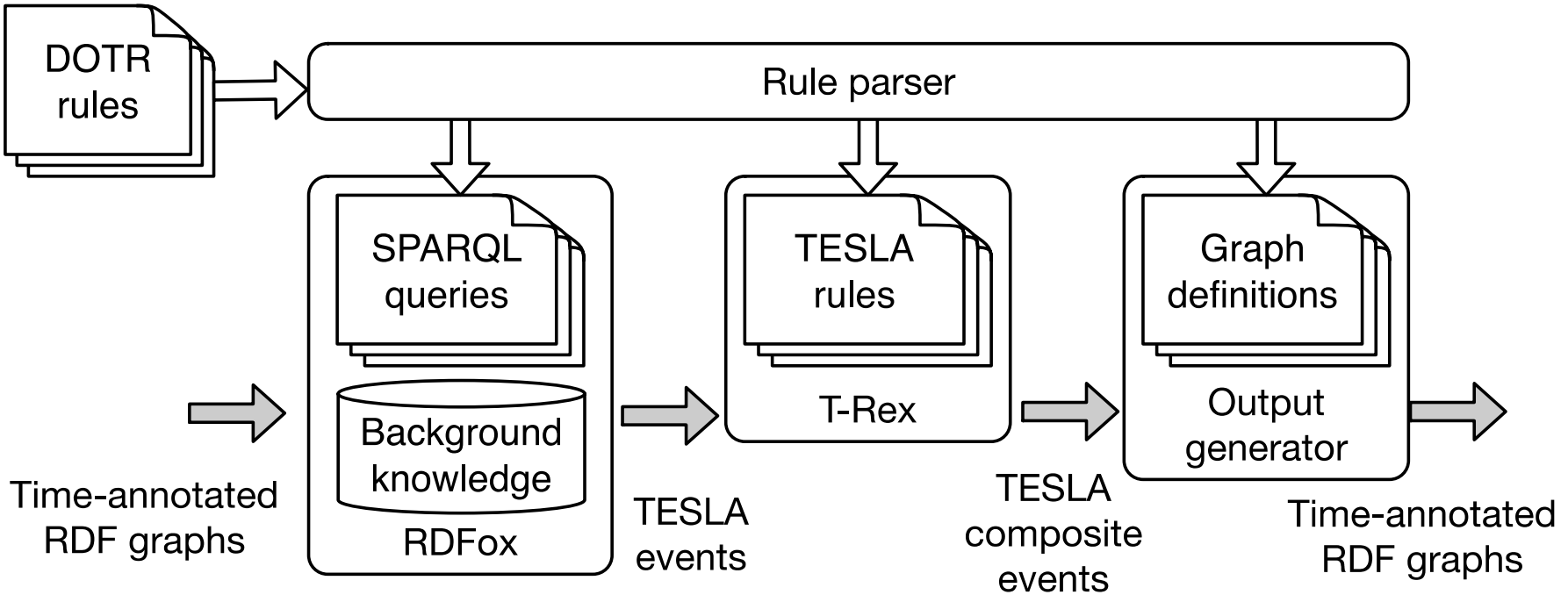
] **and last** HighTemp = [

```
SELECT ?read2 ?room2
WHERE {
        ?read2 rdf:type :Temp. ?read2 :is_from_sensor ?sens2.
        ?sens2 :is_in_room ?room2. ?read2 :has_val ?v.
        FILTER (?v > 60 && ?room2 = Smoke.?room1). }
```

] **within** 5 min **from** Smoke
**with** Fire.?room = Smoke.?room1
**consuming** HighTemp

Event definition
(from streaming and
background knowledge)

# The DOTR implementation



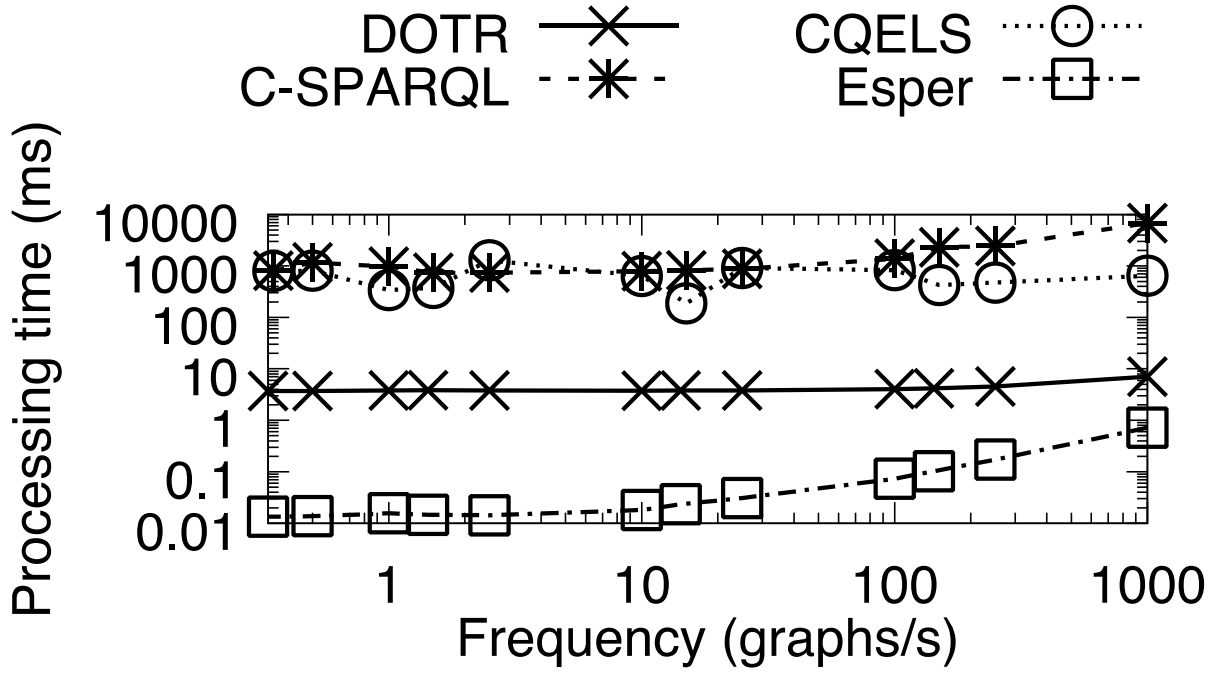
# Evaluation

- Two types of experiments
  - Comparison with state-of-the-art solutions
  - Sensitivity to parameters

# Comparison with state-of-the-art

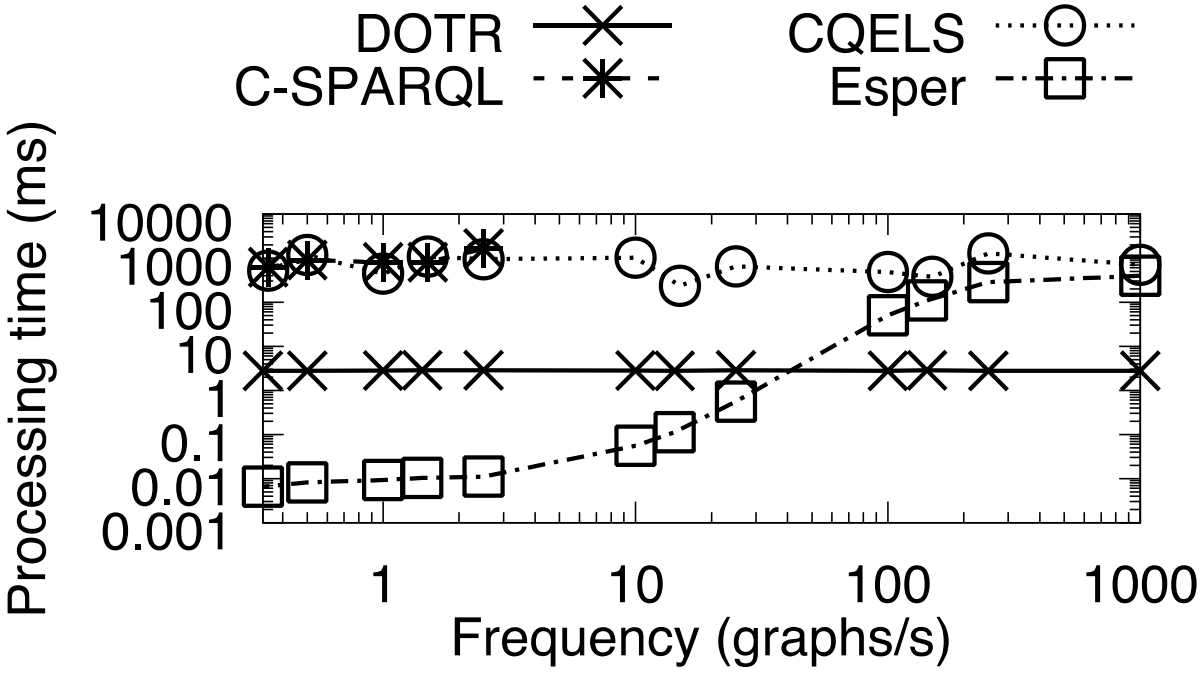
- We compare DOTR with
  - C-SPARQL, CQELS, Esper
- We use CityBench as benchmark
  - Disclaimer: all the CityBench queries are based on windows
  - The same holds for all the other stream reasoning benchmarks ☹

# Comparison with state-of-the-art




- Query Q1: what is the congestion level on each road?
  - Behavior similar to that of C-SPARQL and CQELS
- Ontological reasoning manually encoded in Esper

# Comparison with state-of-the-art



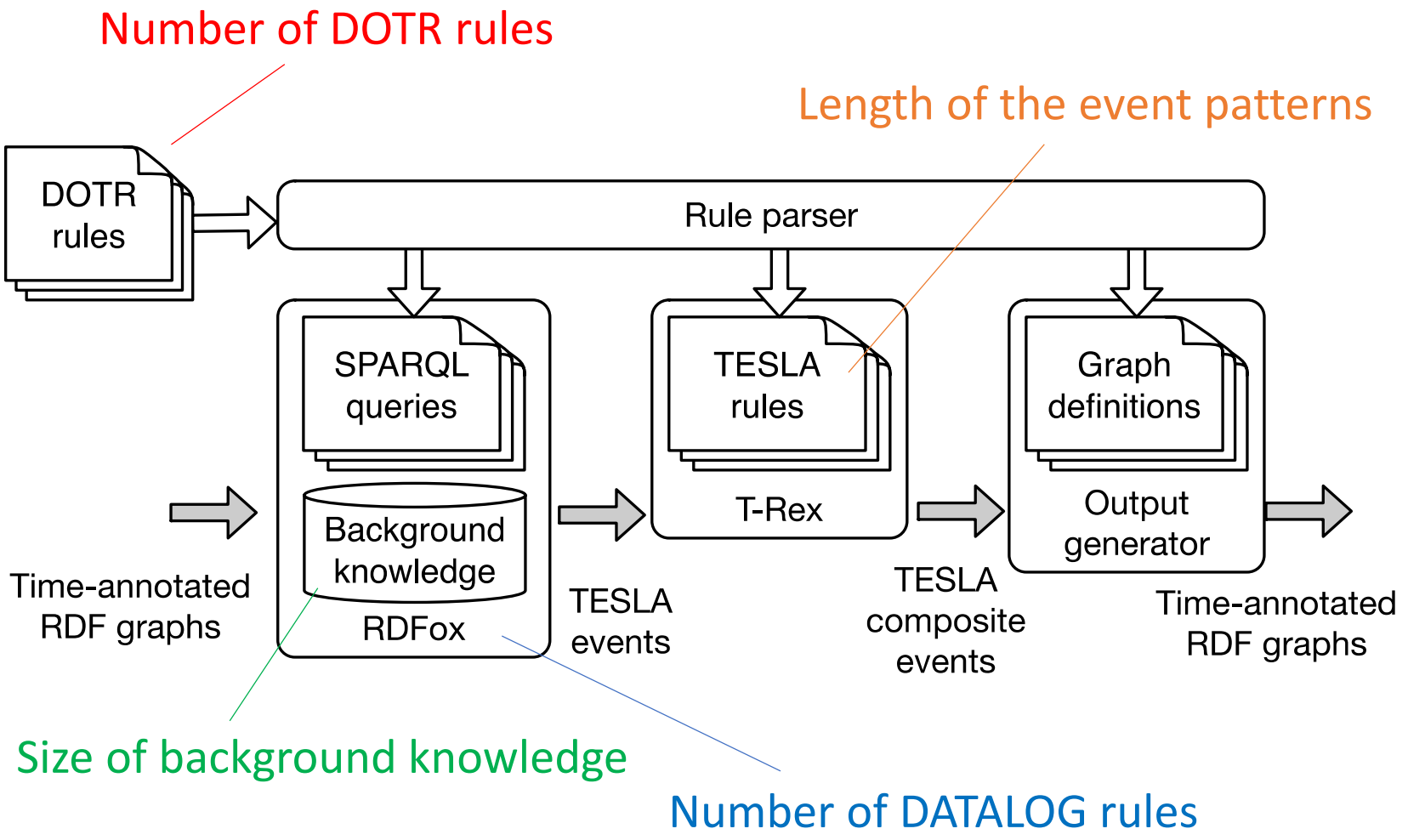

- What is the (*current*) weather condition?
  - Windows force periodic re-evaluation!
  - DOTR avoids duplicate evaluations and detections!

# Sensitivity to parameters

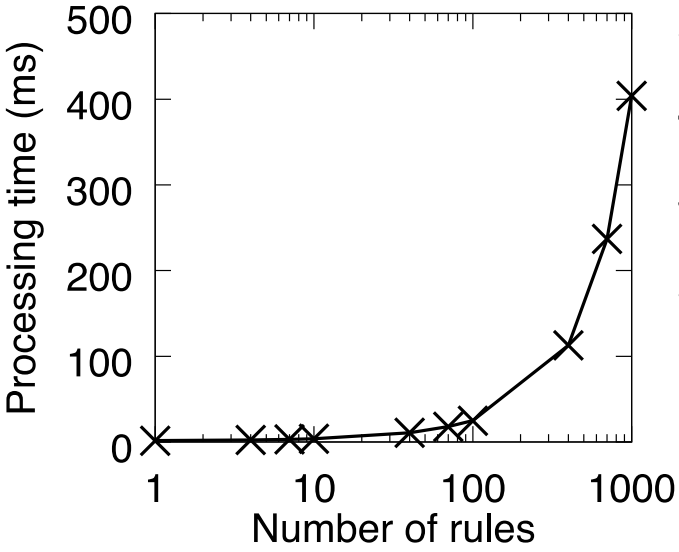
- We set a default scenario (based on synthetic data)
- We study sensitivity by changing one parameter at a time

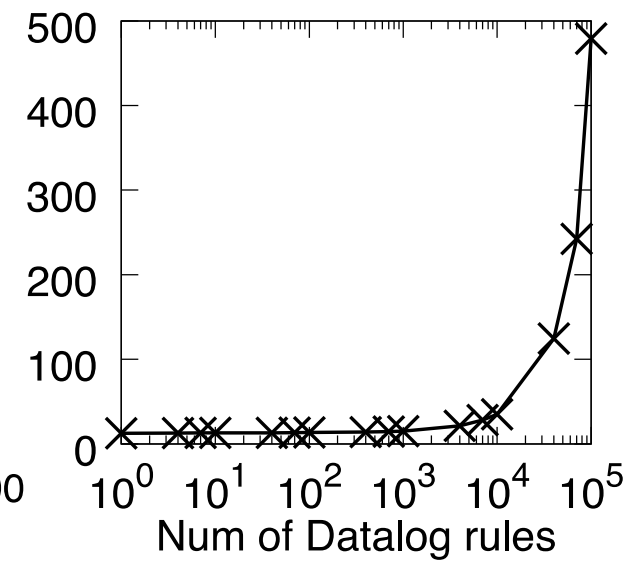
| | Time (ms) | Time (%) |
|---|---|---|
| Knowledge update | 0.20 | 1.65 |
| Ontological reasoning | 1.96 | 16.21 |
| Query evaluation | 7.56 | 62.53 |
| Input translation | 0.19 | 1.57 |
| Event processing | 2.01 | 16.33 |
| Output translation | 0.17 | 1.41 |
| **Total** | **12.09** | **100** |

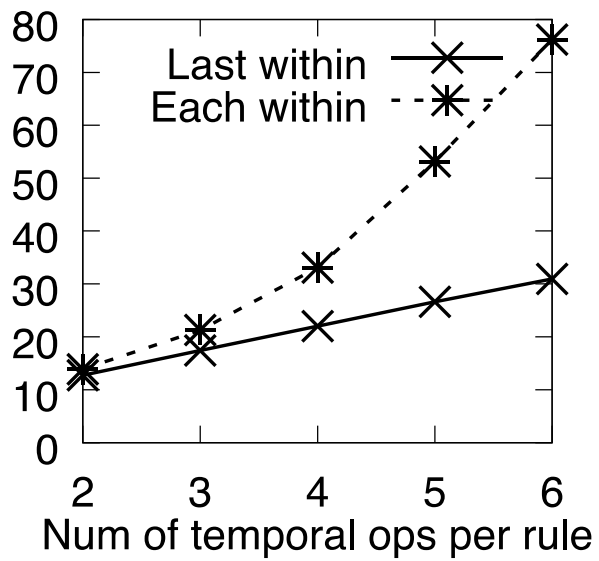
# Parameters under study

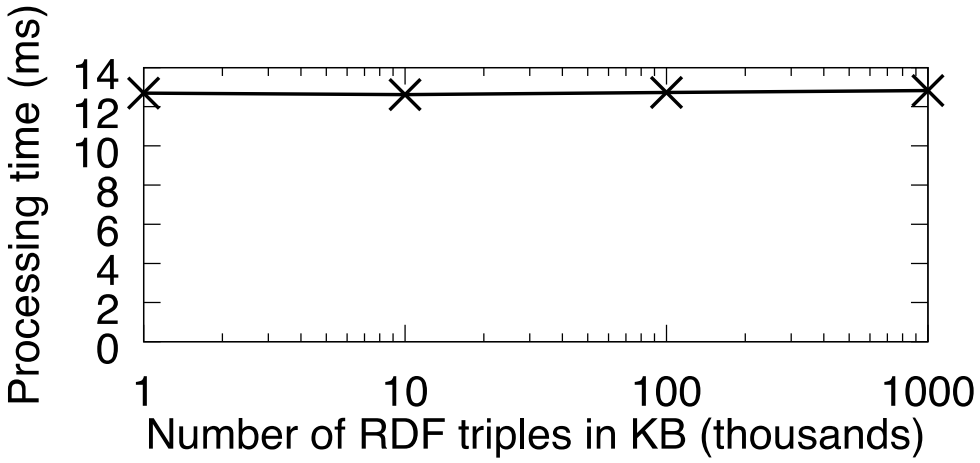


# Sensitivity to parameters



# Conclusions

- DOTR model that decouples ontological and temporal reasoning
- Comparison with state-of-the-art solutions
  - More expressive temporal reasoning
  - Better performance
- Future work
  - Optimized implementation
  - Comparison with other formalisms (ASP)

# Thank you! Questions?

## Efficient Temporal Reasoning on Streams of Events with DOTR

Alessandro Margara, Gianpaolo Cugola

Dario Collavini, Daniele Dell'Aglio