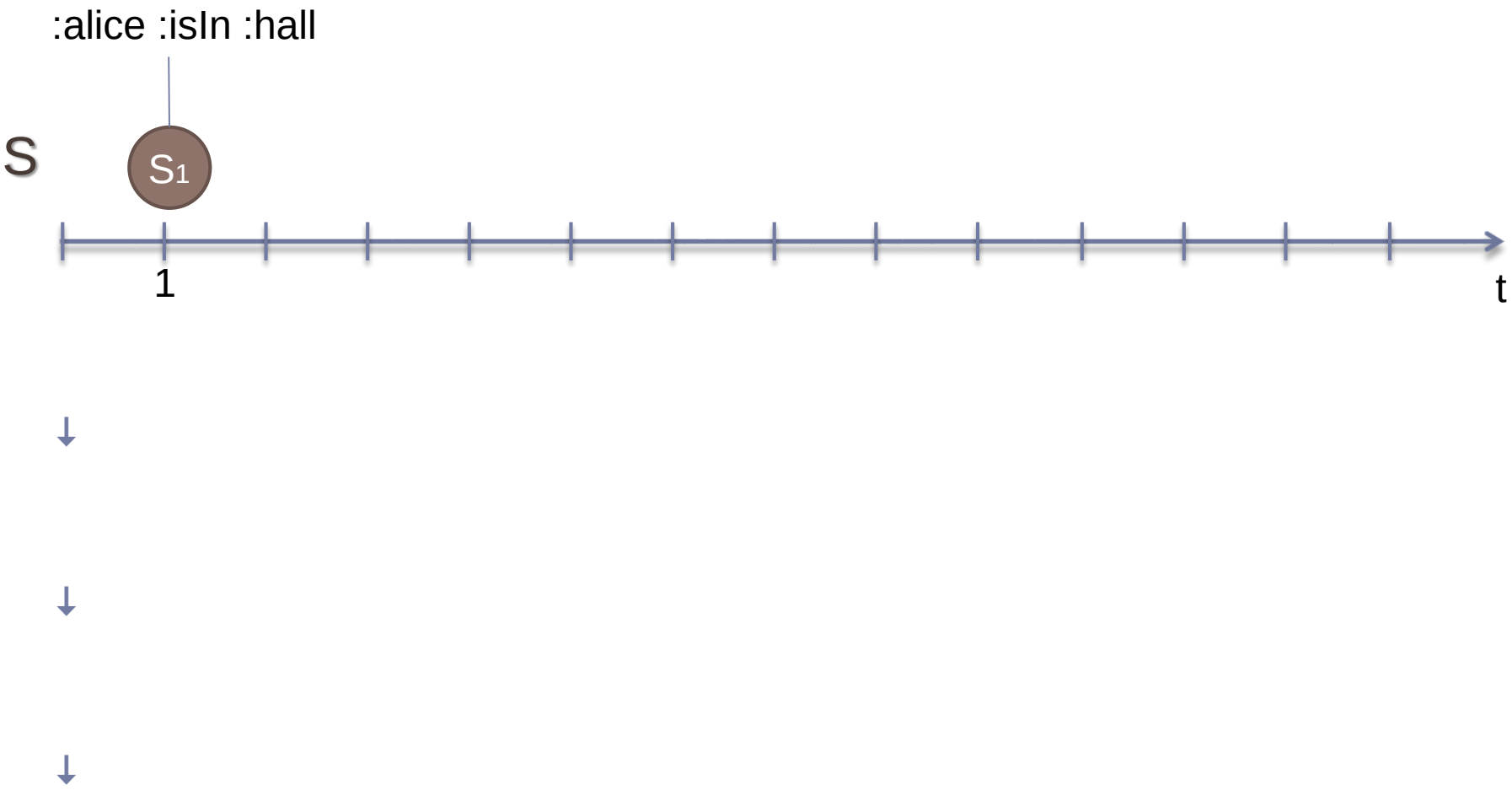
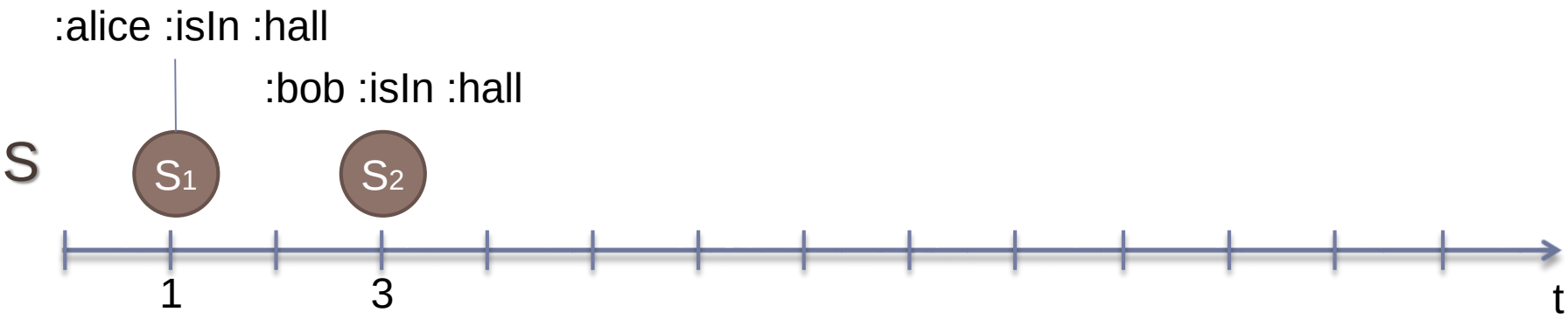
On correctness in RDF stream processor benchmarking

Daniele Dell'Aglio, Jean-Paul Calbimonte, Marco Balduini, Oscar Corcho and Emanuele Della Valle

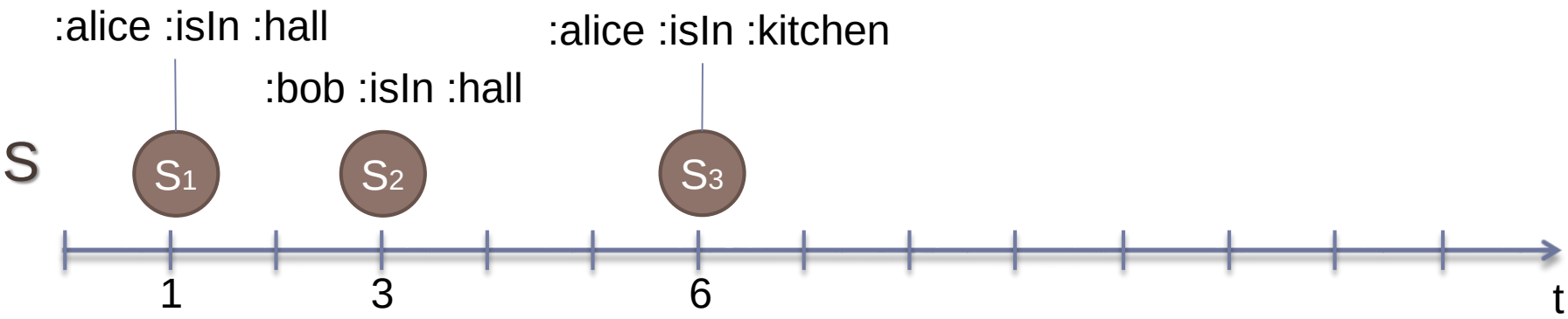
The correctness problem (1)



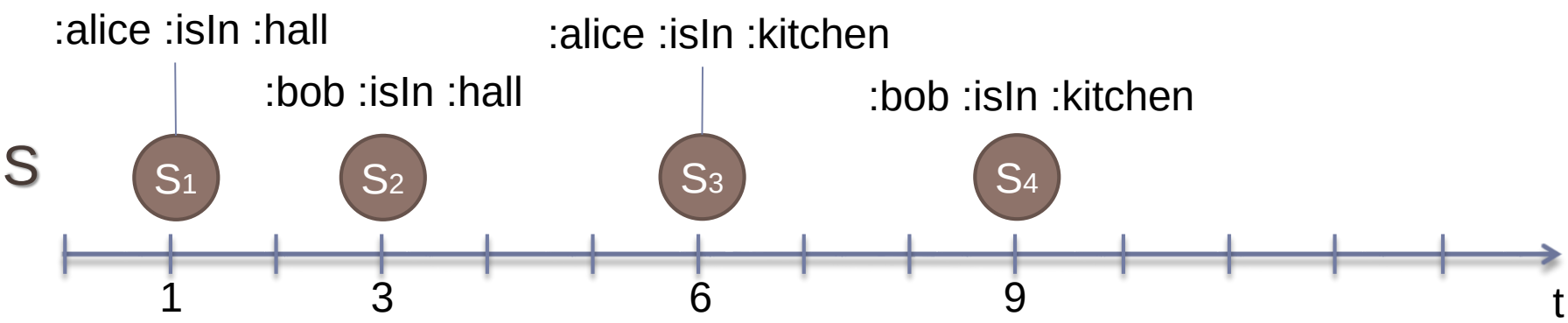
The correctness problem (1)



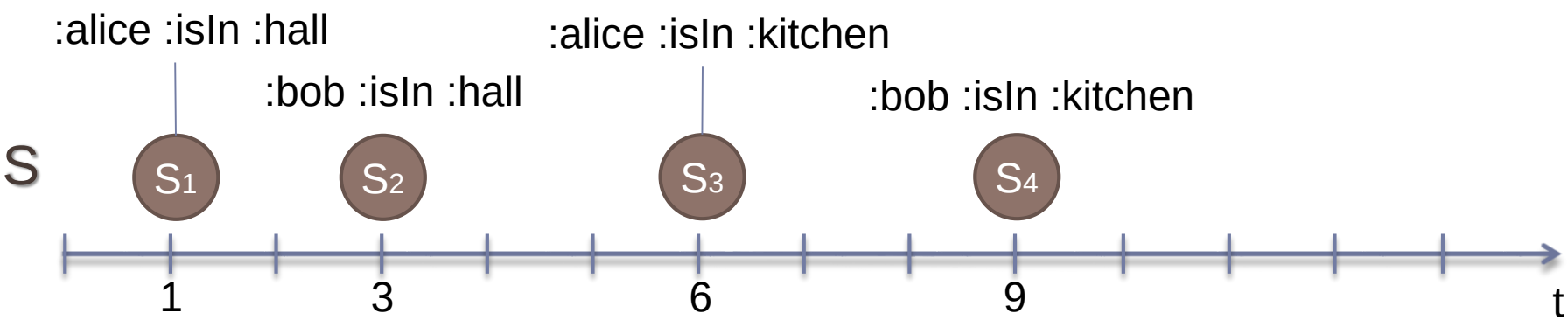
The correctness problem (1)



The correctness problem (1)

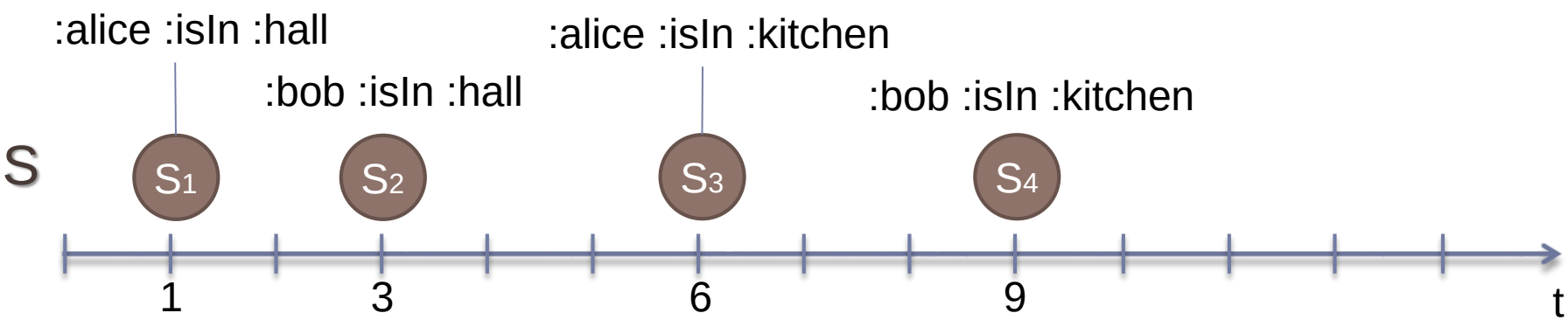


The correctness problem (1)



- Where are Alice and Bob, when they are together?

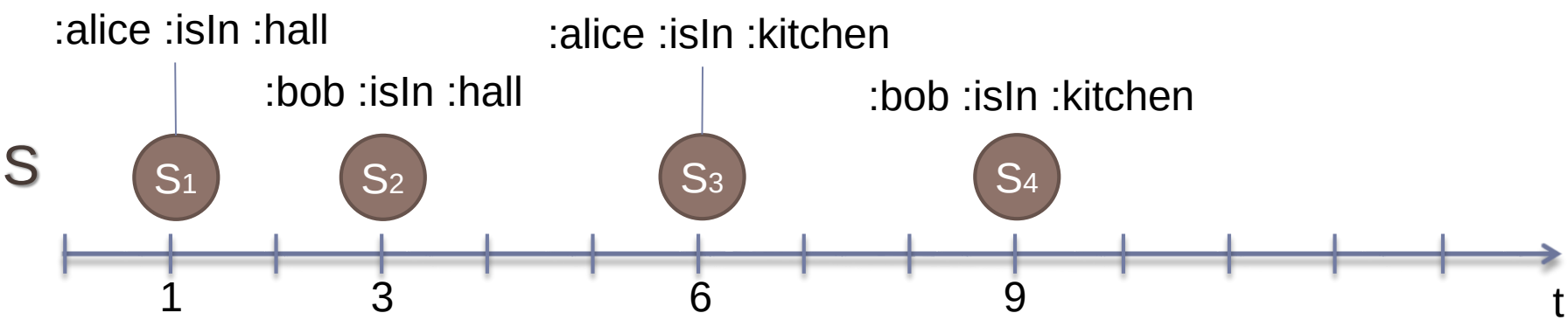
The correctness problem (1)



- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)

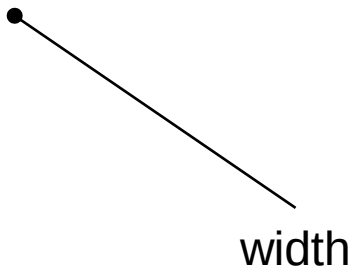
↓

The correctness problem (1)

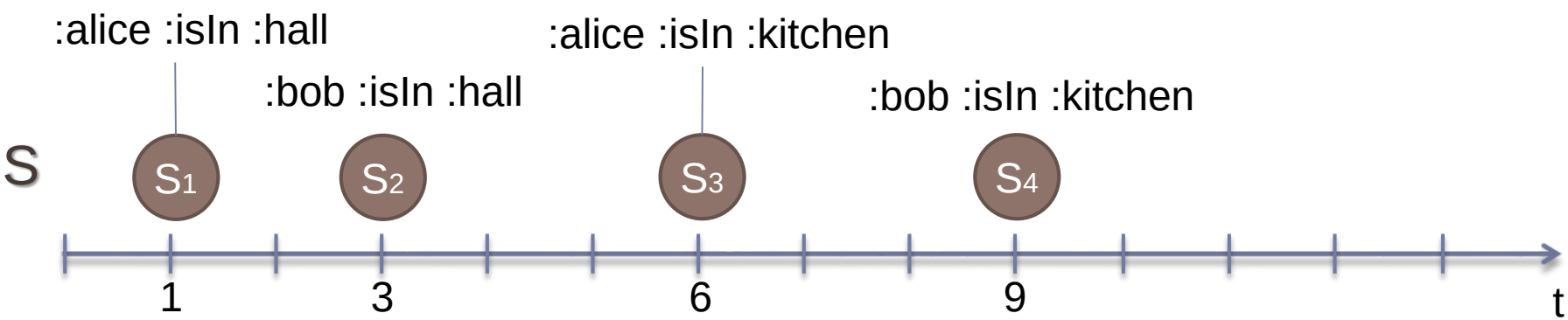


- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)

↓

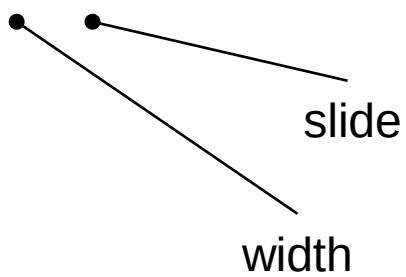


The correctness problem (1)

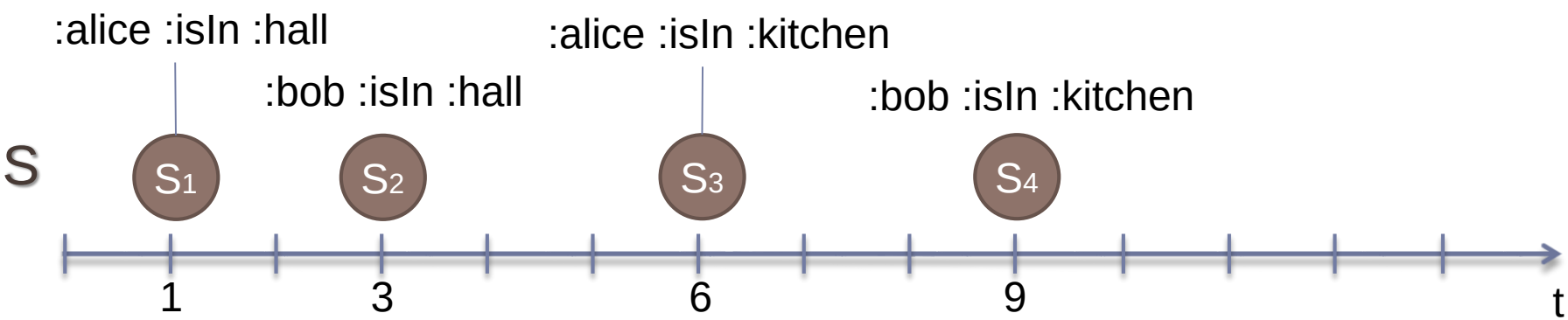


- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)

↓



The correctness problem (1)

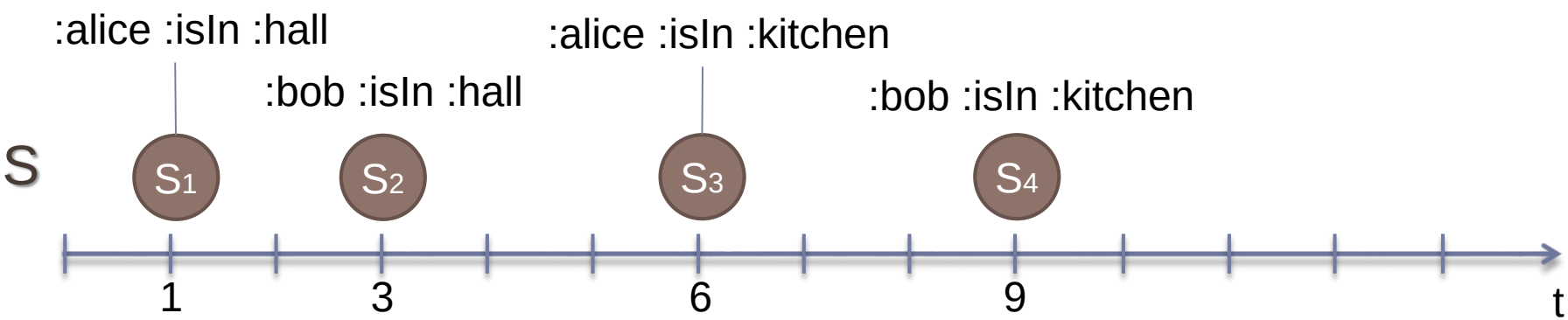


- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)
- Let's execute the experiment 4 times

slide

width

The correctness problem (1)



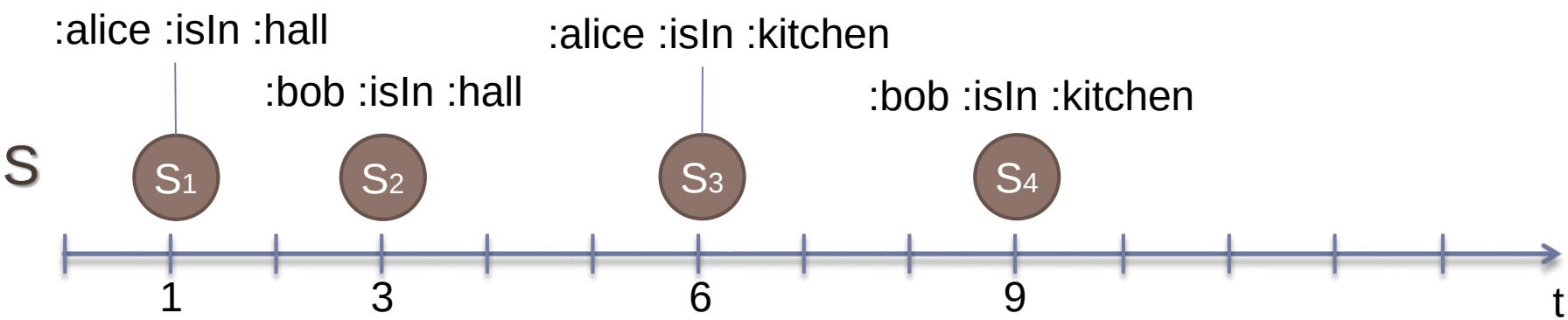
- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)
- Let's execute the experiment 4 times

slide

width

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

The correctness problem (1)



- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)
- Let's execute the experiment 4 times

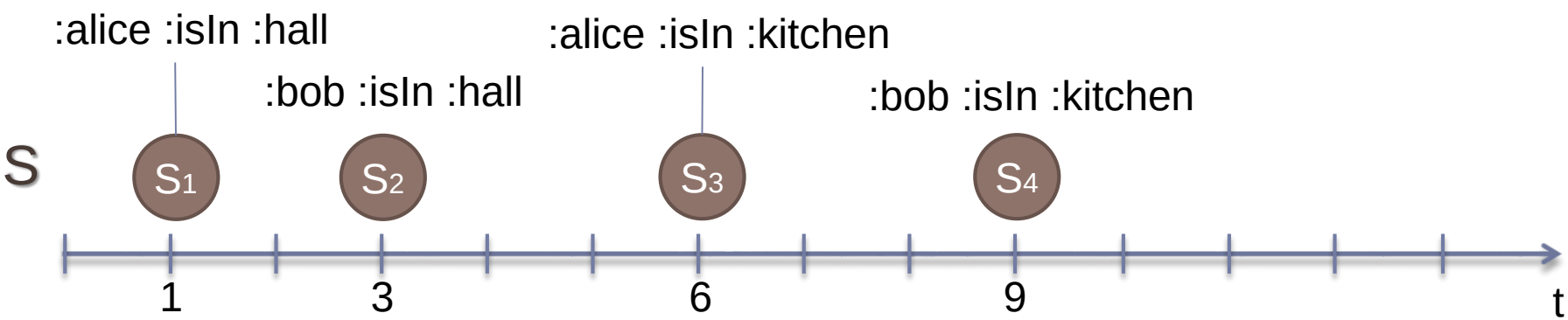
slide

width

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Which is the correct answer?

The correctness problem (1)



- Where are Alice and Bob, when they are together?
- Let's consider a tumbling window W(ω=β=5)
- Let's execute the experiment 4 times

slide

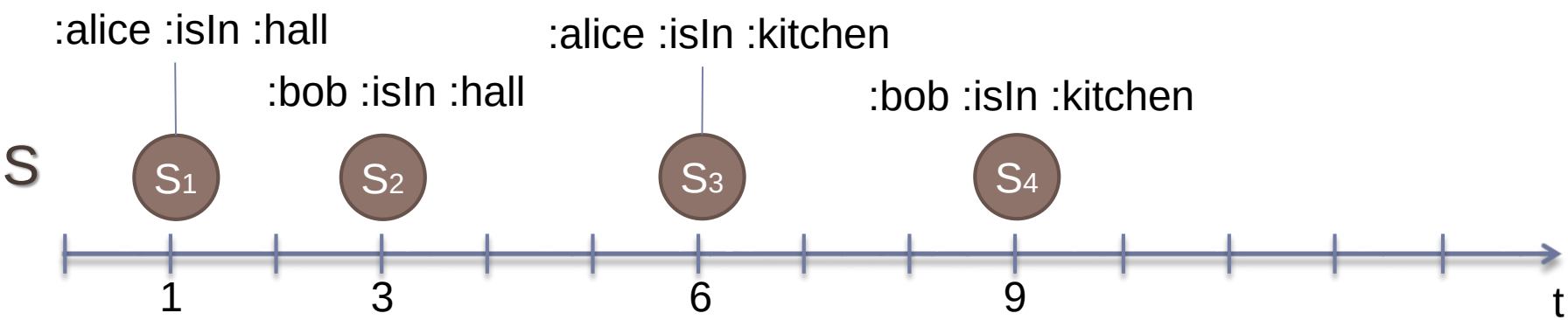
width

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Which is the correct answer?

All of them!

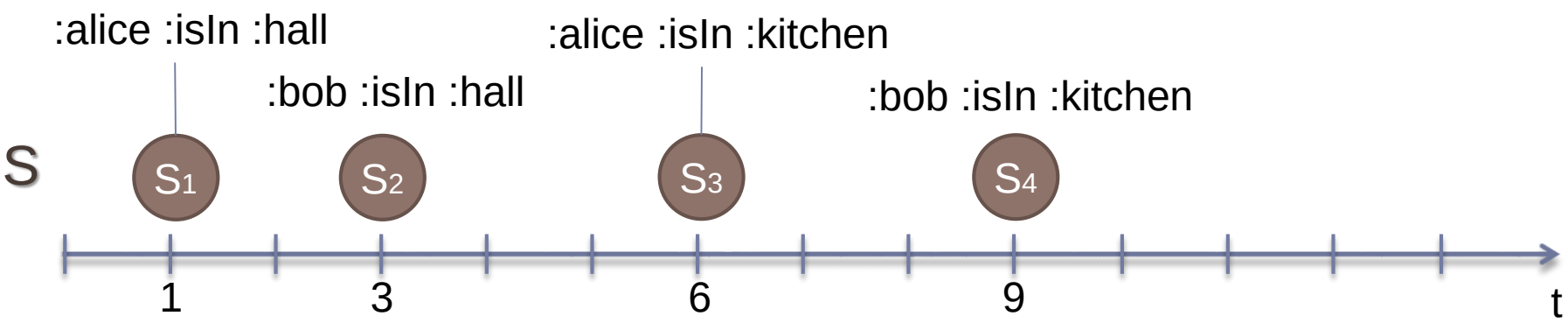
The correctness problem (2)



System 1

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

The correctness problem (2)



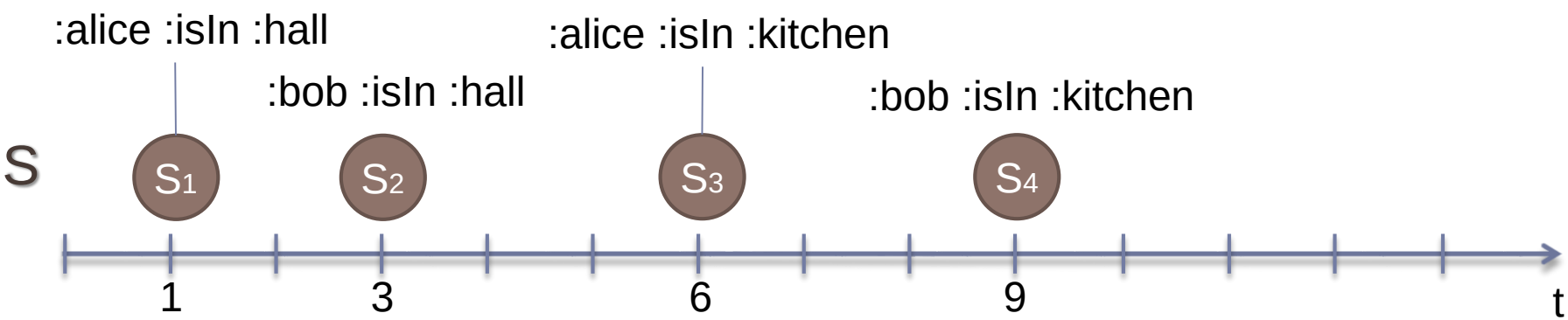
System 1

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

The correctness problem (2)



System 1

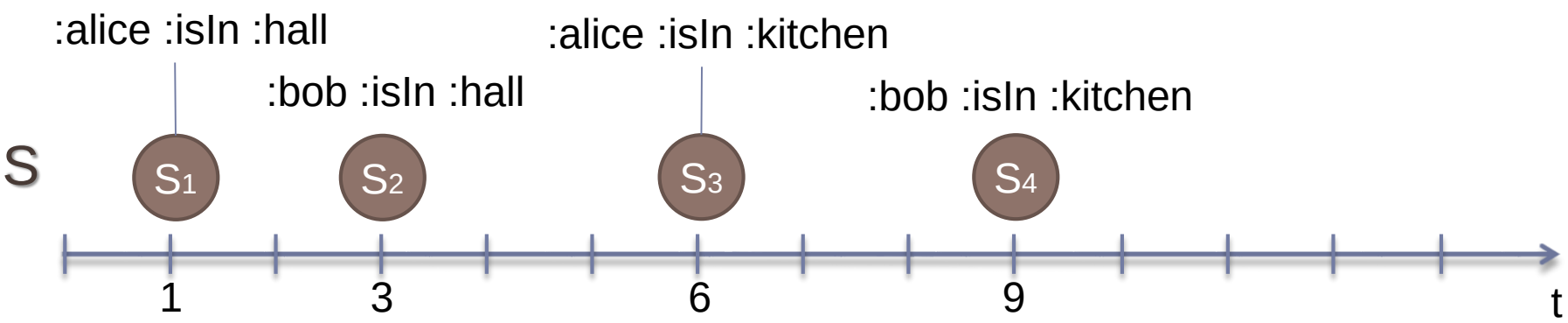
Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

Which system behaves in the correct way?

The correctness problem (2)



System 1

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

Which system behaves in the correct way?

Both!

Different results for the same query? Yes!

Different results for the same query? Yes!

- Given the same data and the same query, we notice that **a system** can provide **multiple correct answers**
  - The operational semantics of the systems often do not explain this behaviour

Different results for the same query? Yes!

- Given the same data and the same query, we notice that **a system** can provide **multiple correct answers**
  - The operational semantics of the systems often do not explain this behaviour

- **Two systems** can provide **different correct answers**
  - Even if the system implements similar operational semantics, they behave in different ways (that the model does not explain)

Different results for the same query? Yes!

- Given the same data and the same query, we notice that **a system** can provide **multiple correct answers**
  - The operational semantics of the systems often do not explain this behaviour

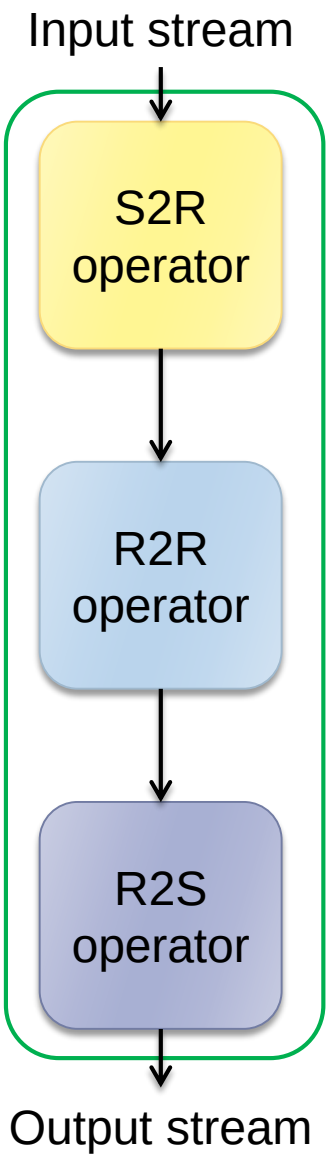
- **Two systems** can provide **different correct answers**
  - Even if the system implements similar operational semantics, they behave in different ways (that the model does not explain)

- Why is it important to understand those behaviours?
  - To assess the correct implementation of the systems
  - To improve the comprehension of the benchmarking

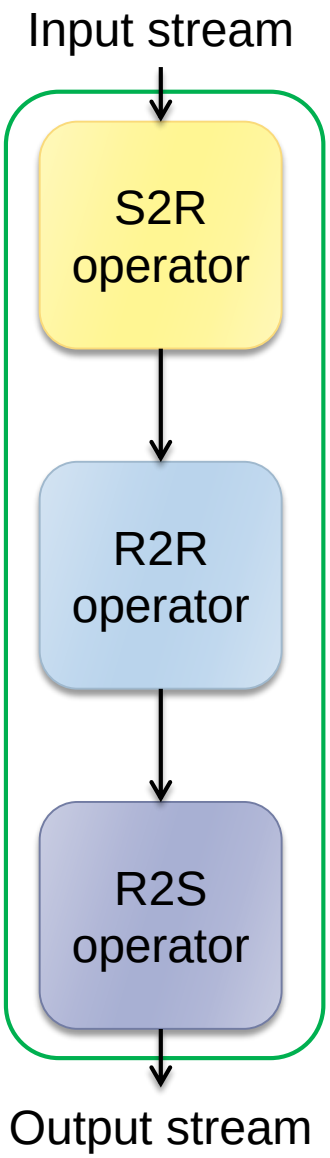
Our contribution

- A common **model** for the RDF stream processor operational semantics
- We propose **CSR-bench**, an extension of the SRbench benchmark that focuses on correctness
  - An **oracle** (an automatic correctness validator)
  - A **test** suite
- We considered the window-based RDF stream processors
  - CQELS
  - C-SPARQL
  - $\text{SPARQL}_{\text{stream}}$

The CSR Model (by CQL)

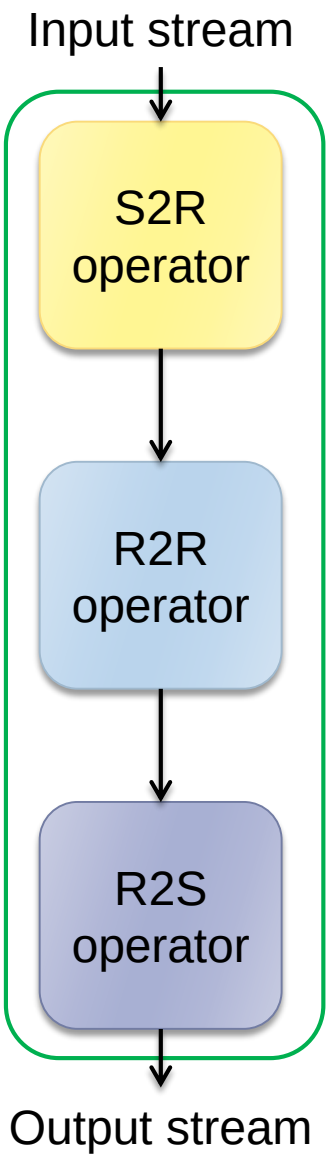


The CSR Model (by CQL)



- Converts the **infinite** stream of RDF elements in a **finite** set of mappings
- The window operators: time-based, tuple-based, …

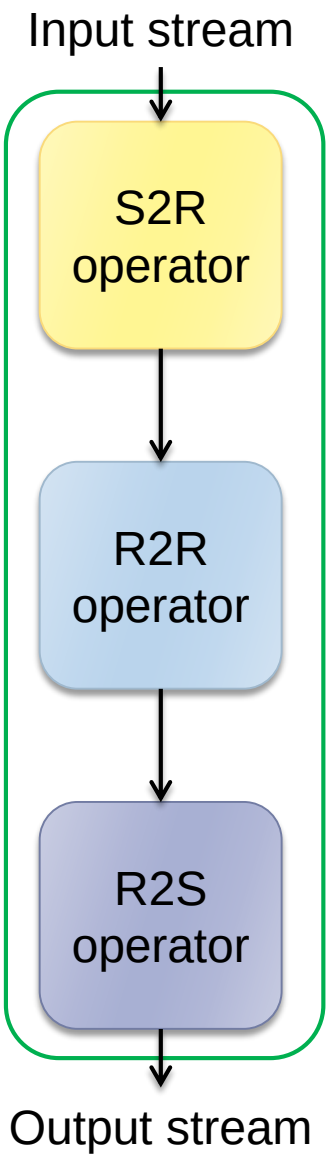
The CSR Model (by CQL)



- Converts the **infinite** stream of RDF elements in a **finite** set of mappings
- The window operators: time-based, tuple-based, …

- Transform a set of mappings in another set of mappings
- **SPARQL** 1.0/1.1 queries

The CSR Model (by CQL)

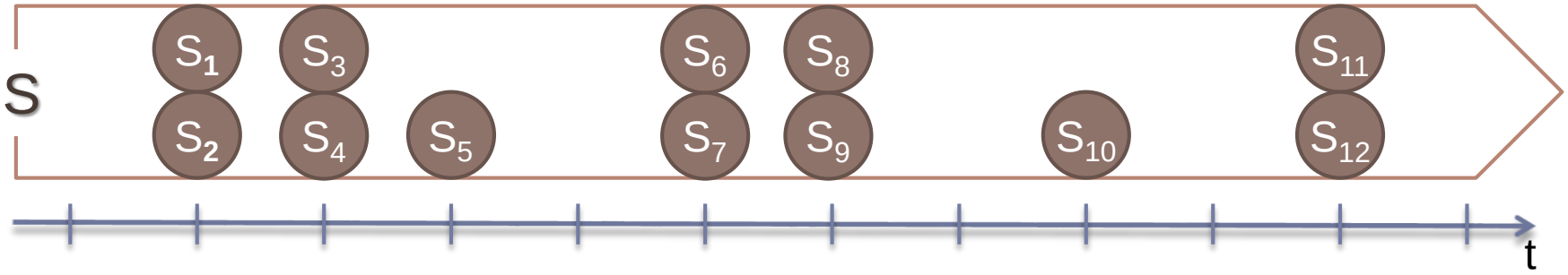


- Converts the **infinite** stream of RDF elements in a **finite** set of mappings
- The window operators: time-based, tuple-based, …

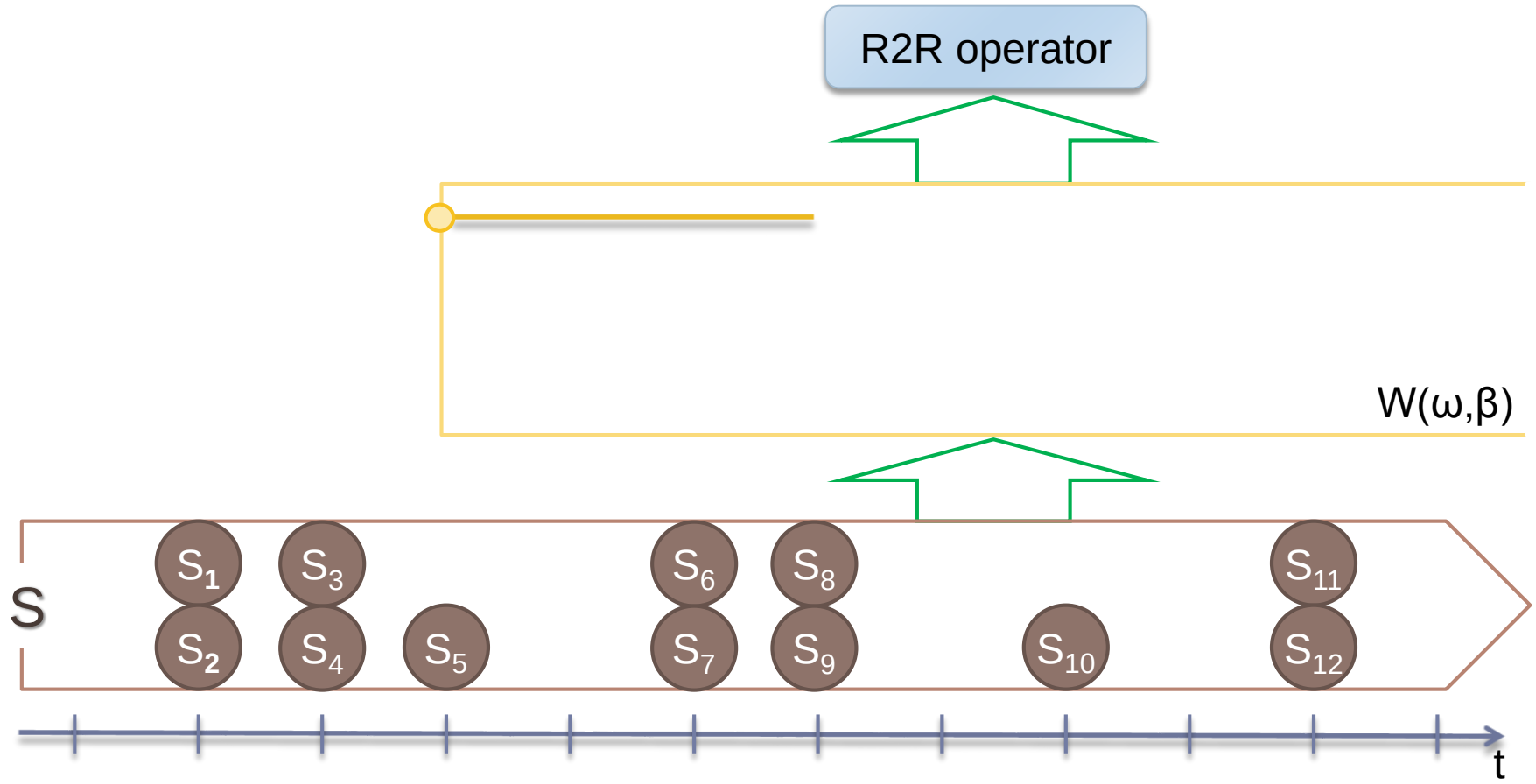
- Transform a set of mappings in another set of mappings
- **SPARQL** 1.0/1.1 queries

- Each set of mapping produced by the R2R operator is **transformed and appended** to the output stream
- Operators: RStream, DStream, IStream

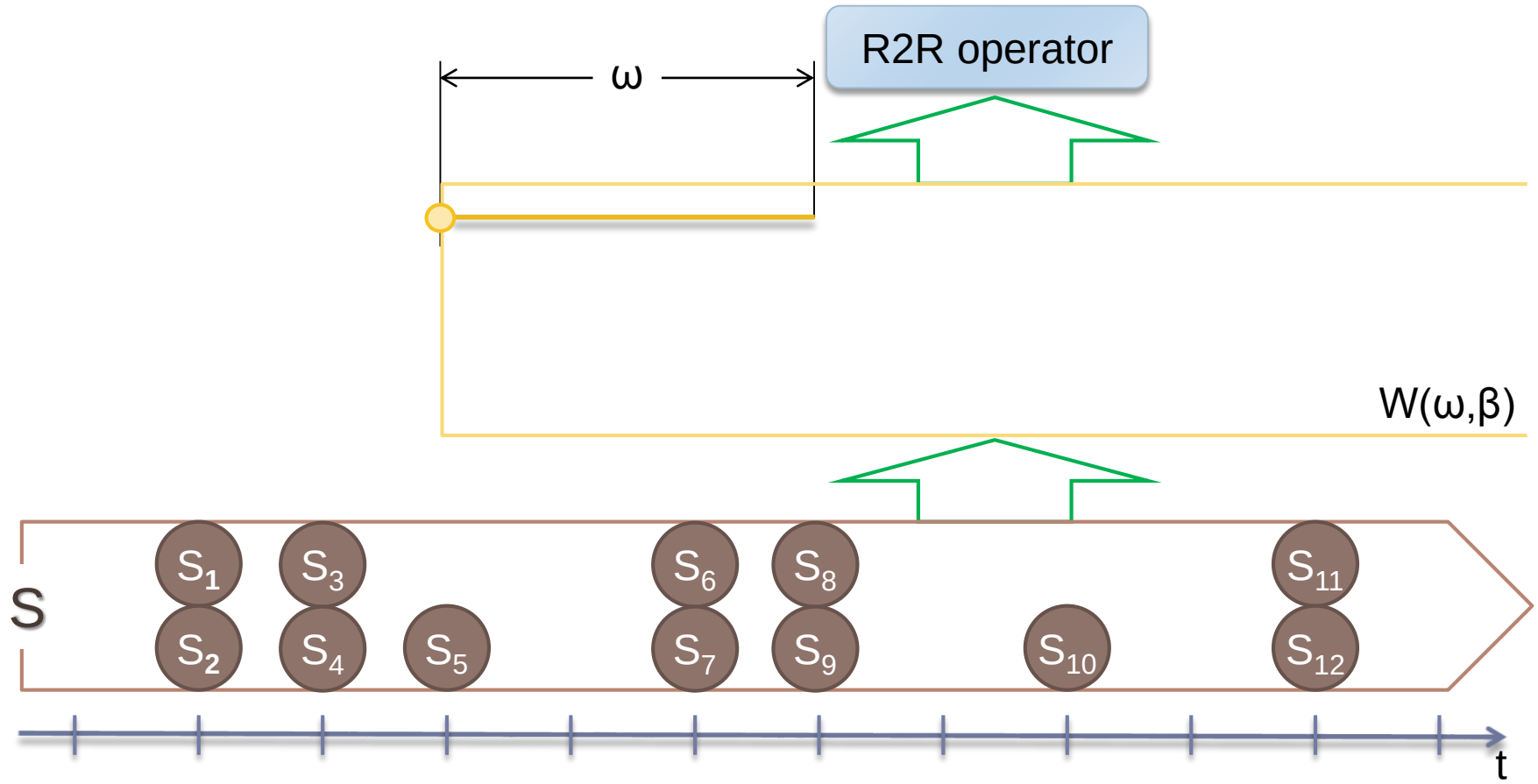
The window operator (through SECRET)



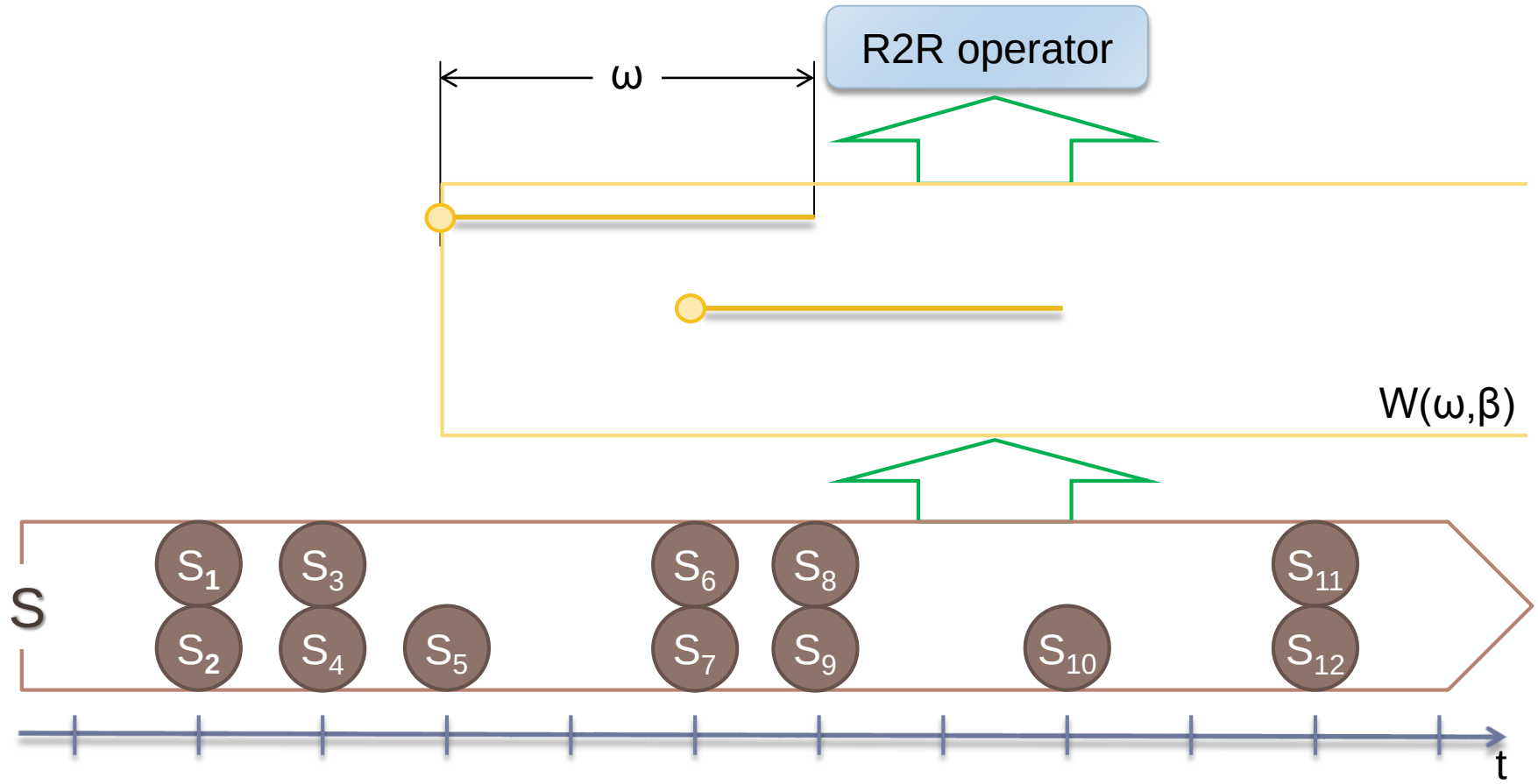
The window operator (through SECRET)



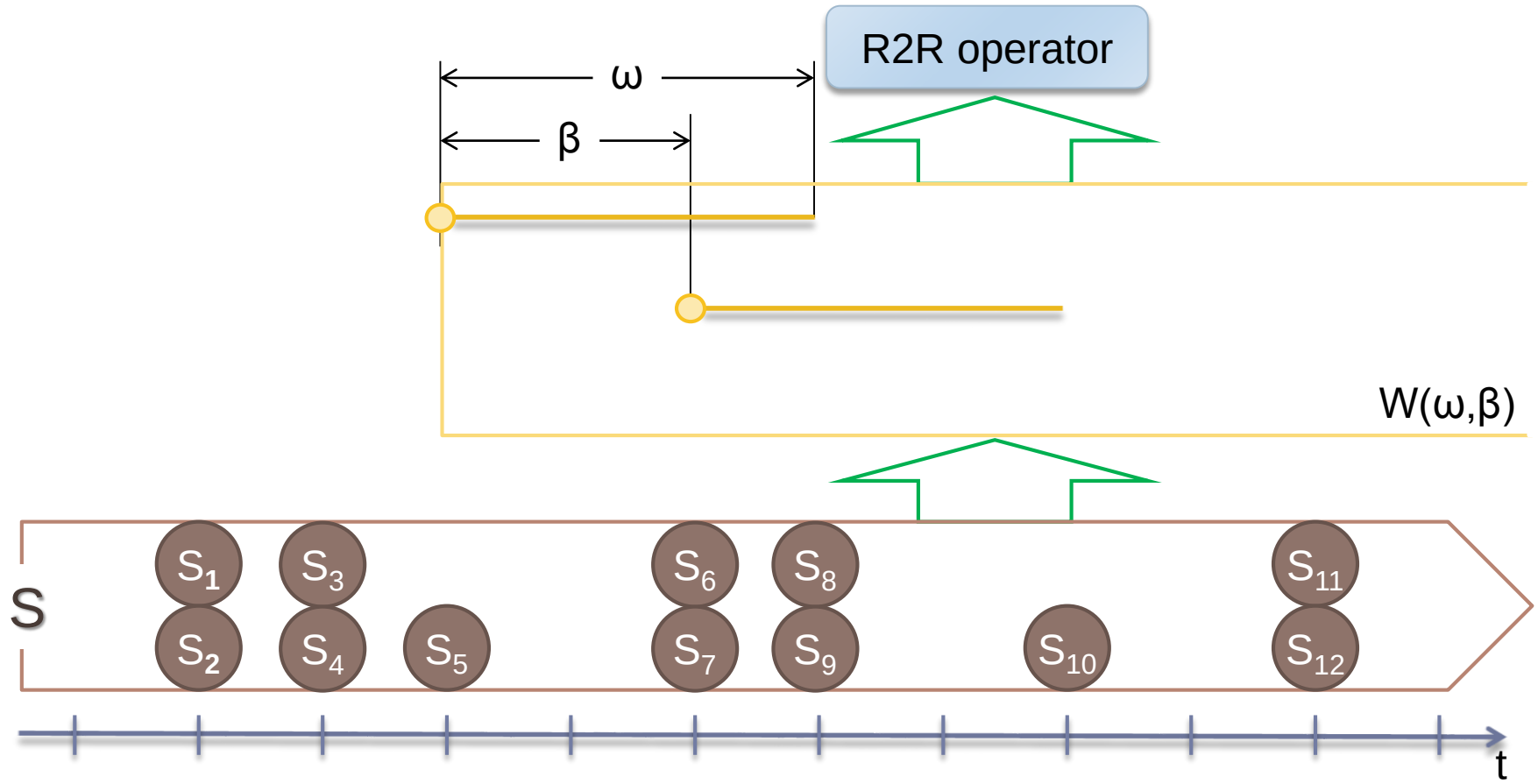
The window operator (through SECRET)



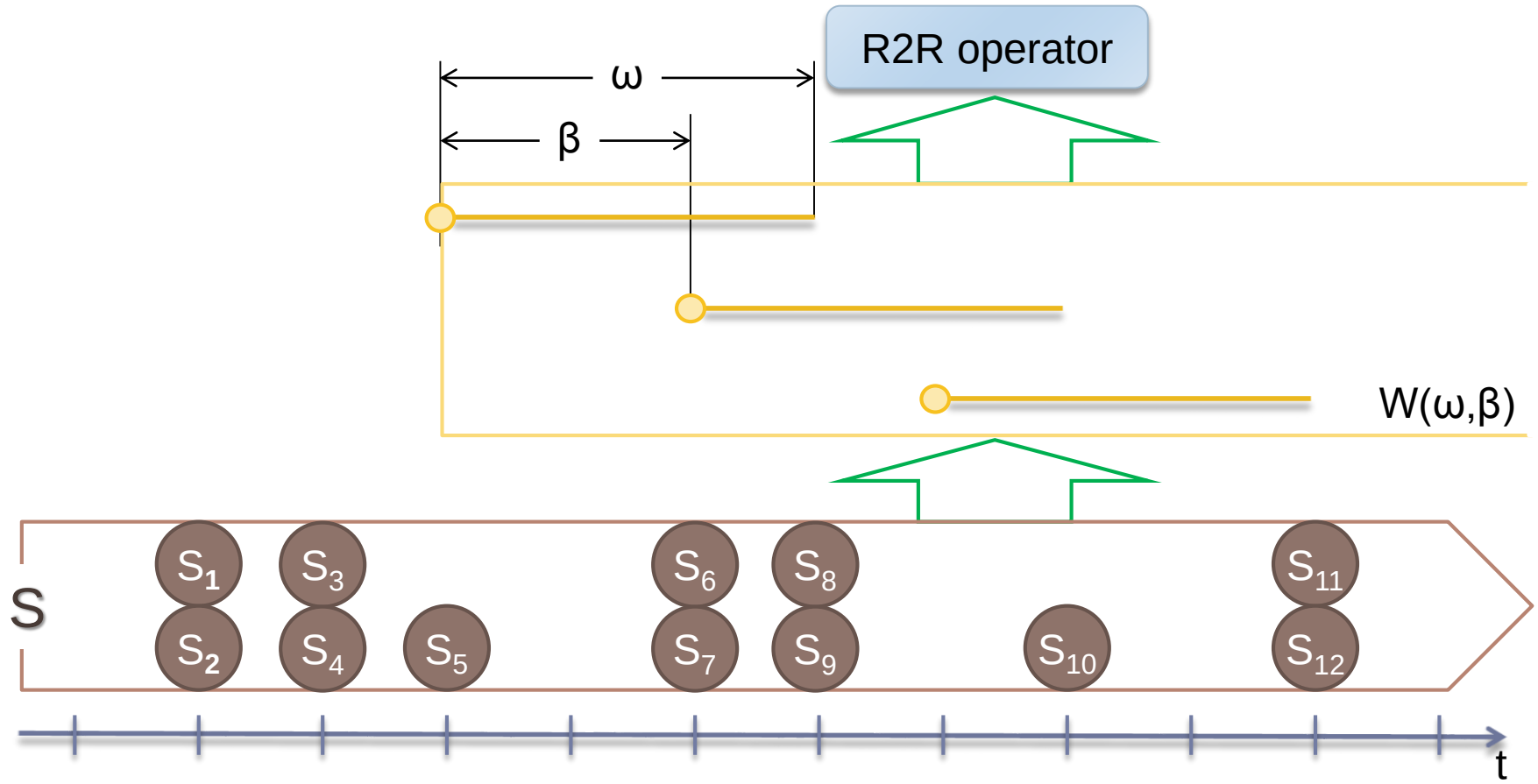
The window operator (through SECRET)



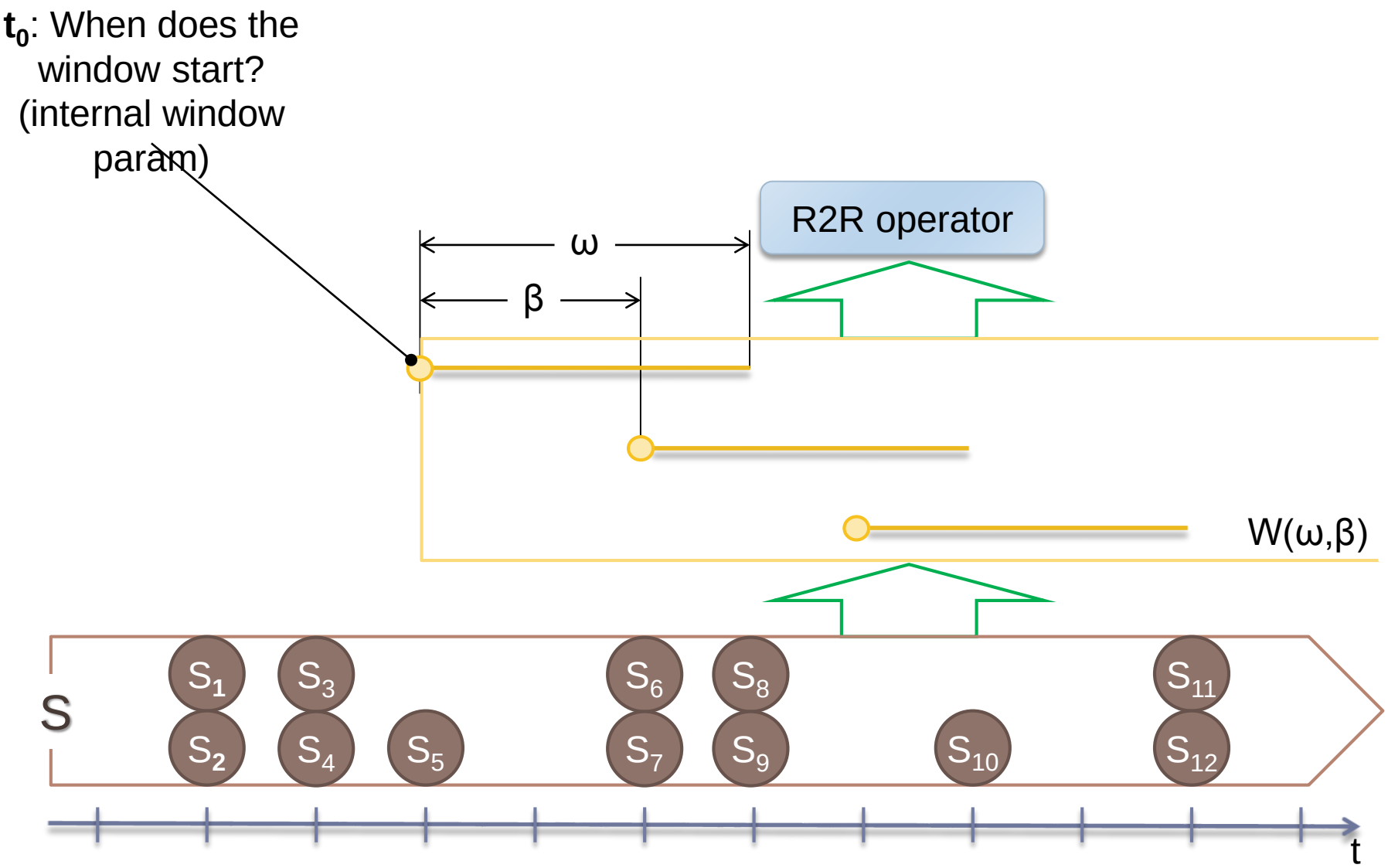
The window operator (through SECRET)



The window operator (through SECRET)



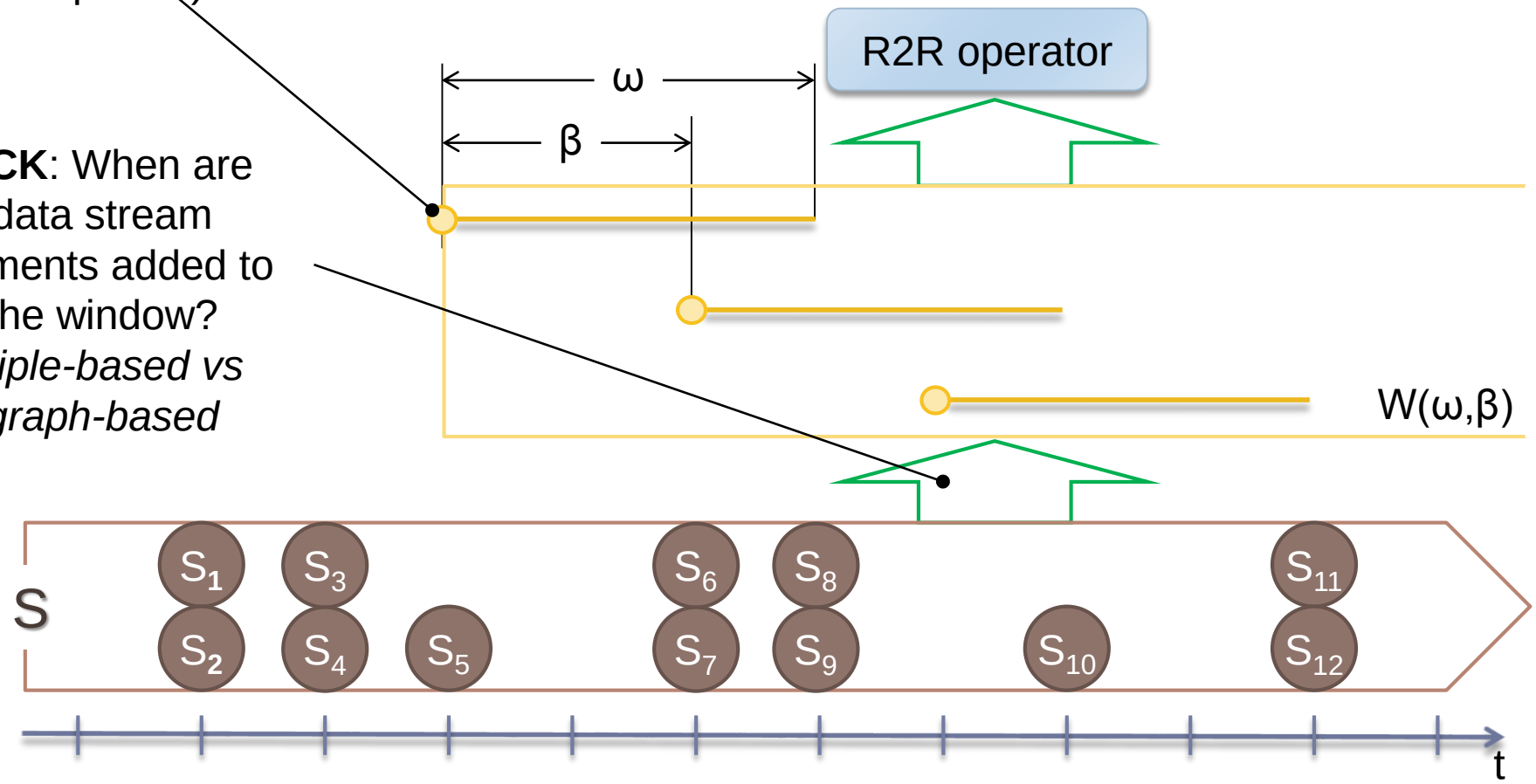
The window operator (through SECRET)



The window operator (through SECRET)

t_0: When does the window start? (internal window param)

TICK: When are data stream elements added to the window? *Triple-based vs graph-based*

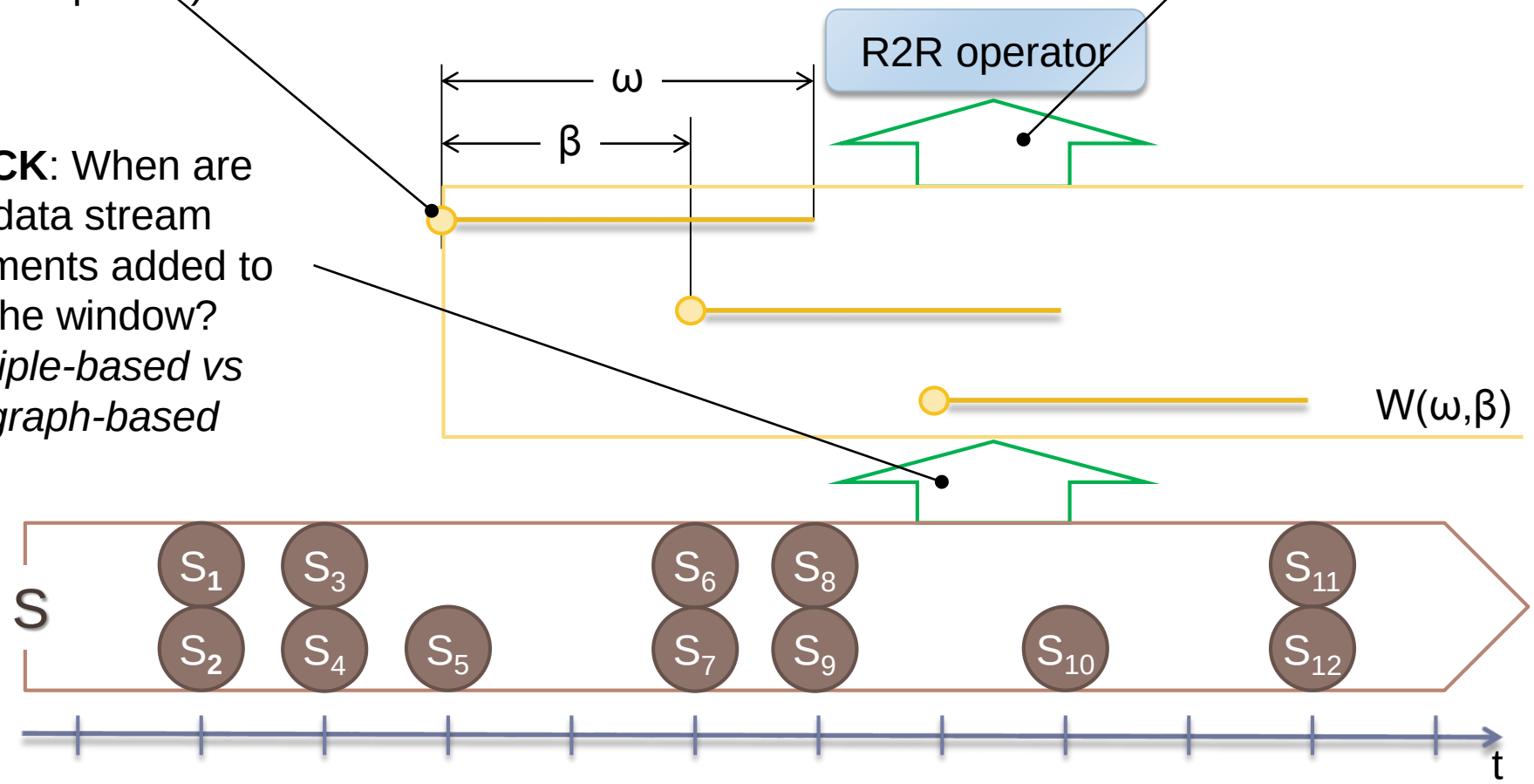


The window operator (through SECRET)

t_0: When does the window start? (internal window param)

REPORT: When is the window content made available to the R2R operator? *Non-empty content, Content-change, Window-close, Periodic*

TICK: When are data stream elements added to the window? *Triple-based vs graph-based*

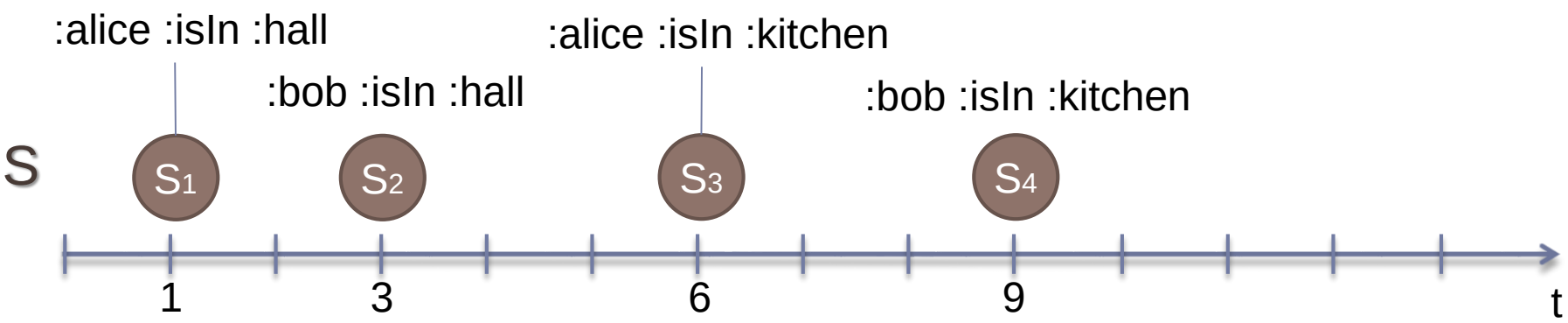


Classification of existing systems

- The report and the tick policies are related to the RSPs implementation and on how they implement the window operators
- Analysing the RSPs and their related documentation (papers, technical reports, etc.), we classified the systems:

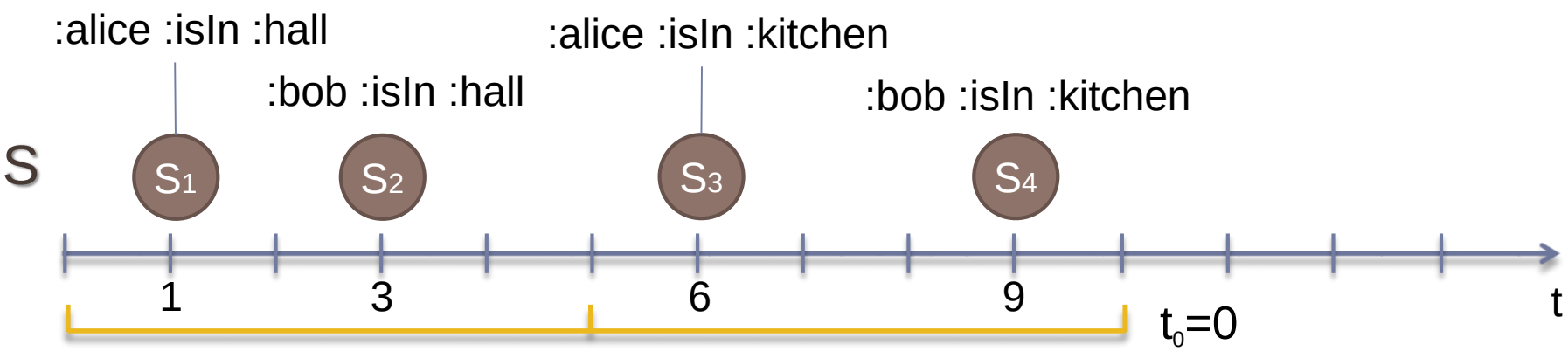
	CQELS	C-SPARQL	$SPARQL_{stream}$
Report	Content-change	Window-close Non-empty content	Window-close Non-empty content
Tick	Tuple-driven	Tuple-driven	Tuple-driven
Empty relation notification	No	Yes	No

RSP output correctness



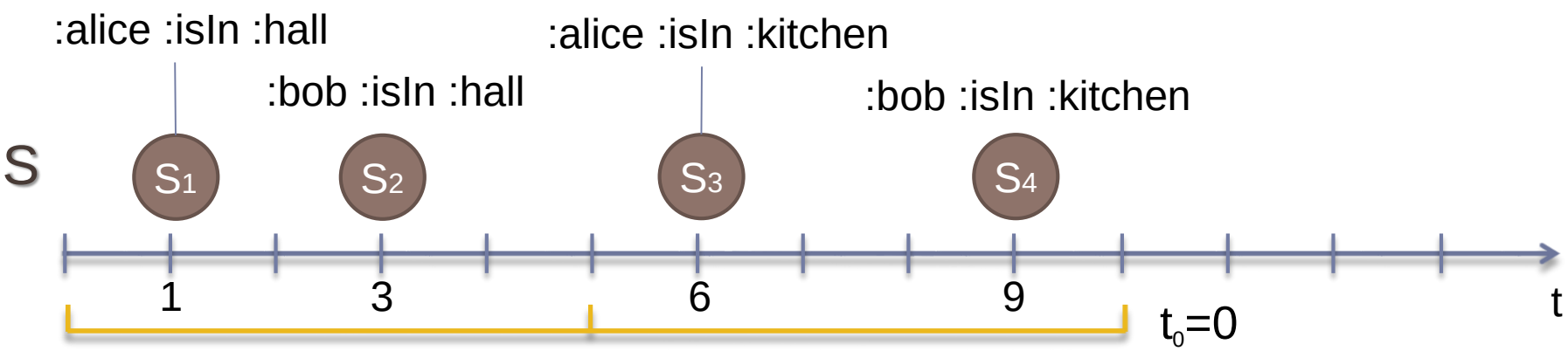
Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

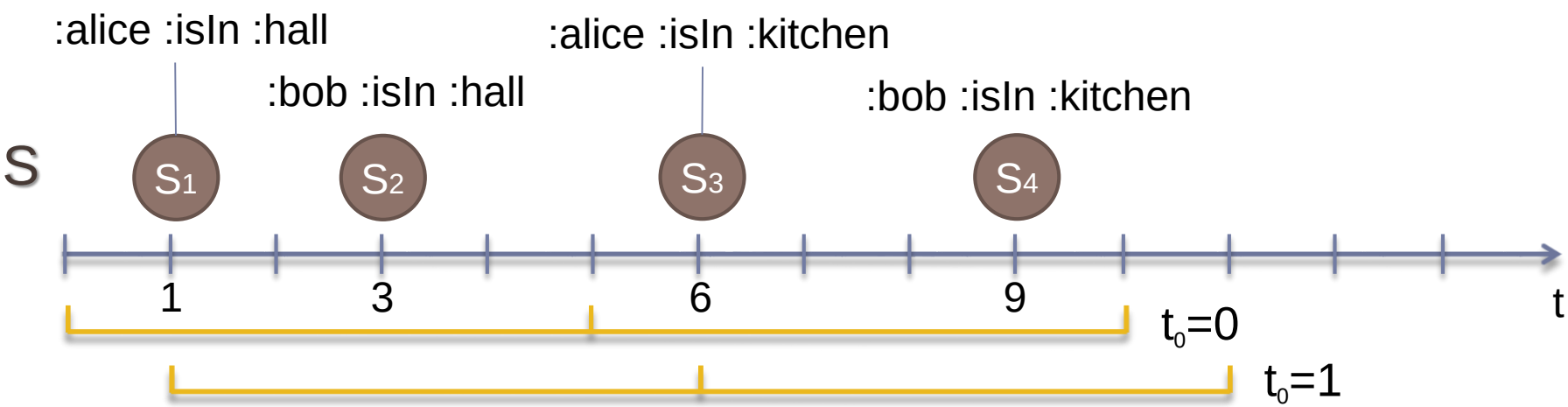
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]

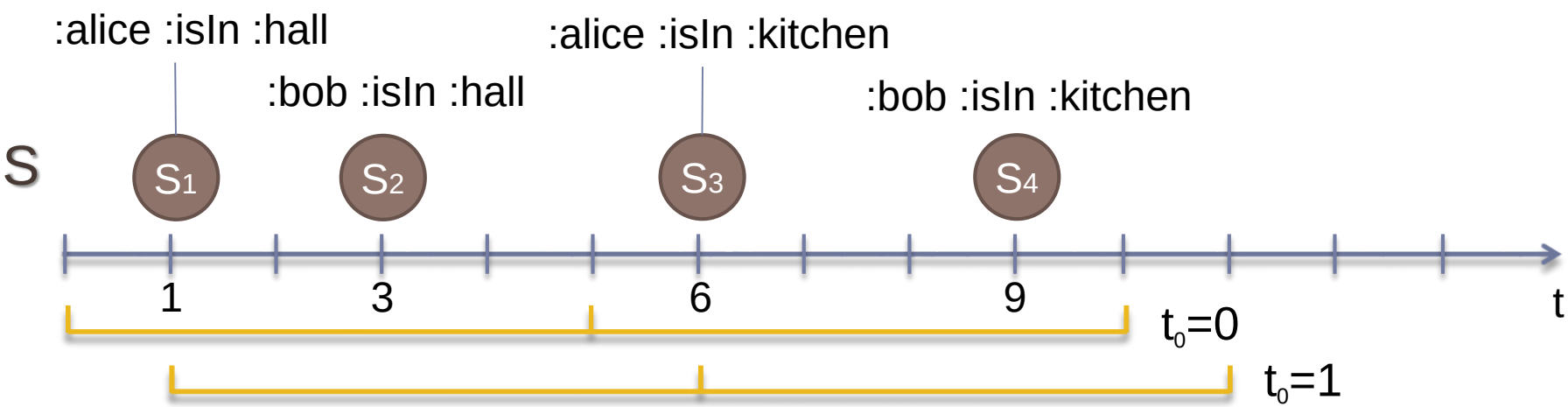
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]

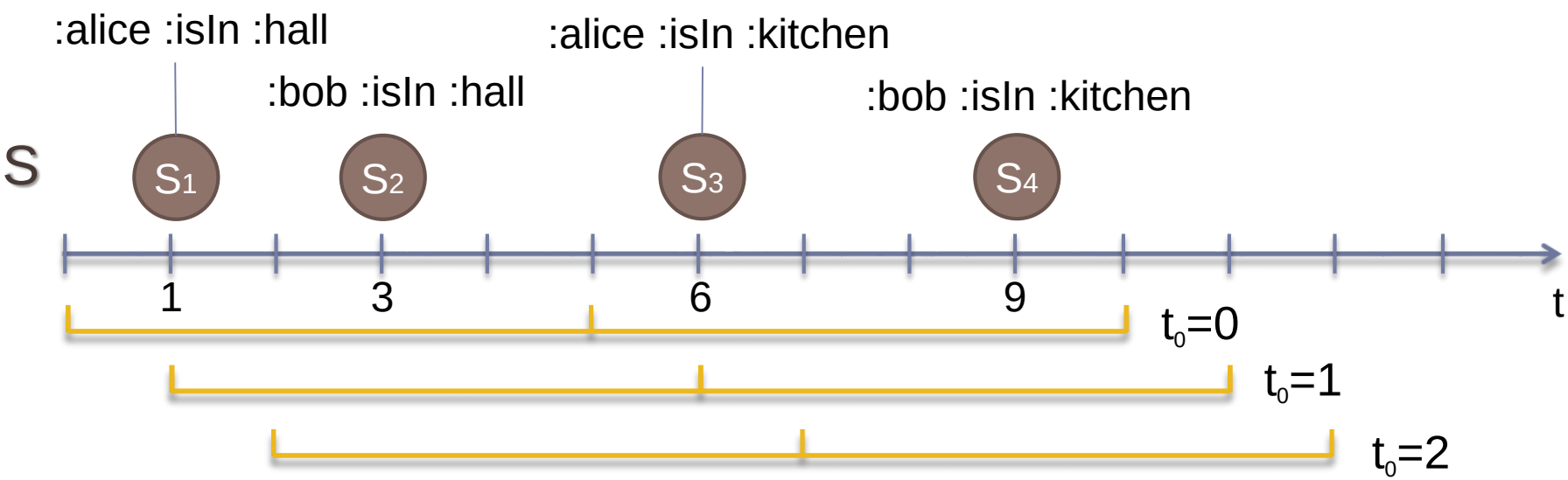
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]
t_0=1	:hall [6]	:kitchen [11]

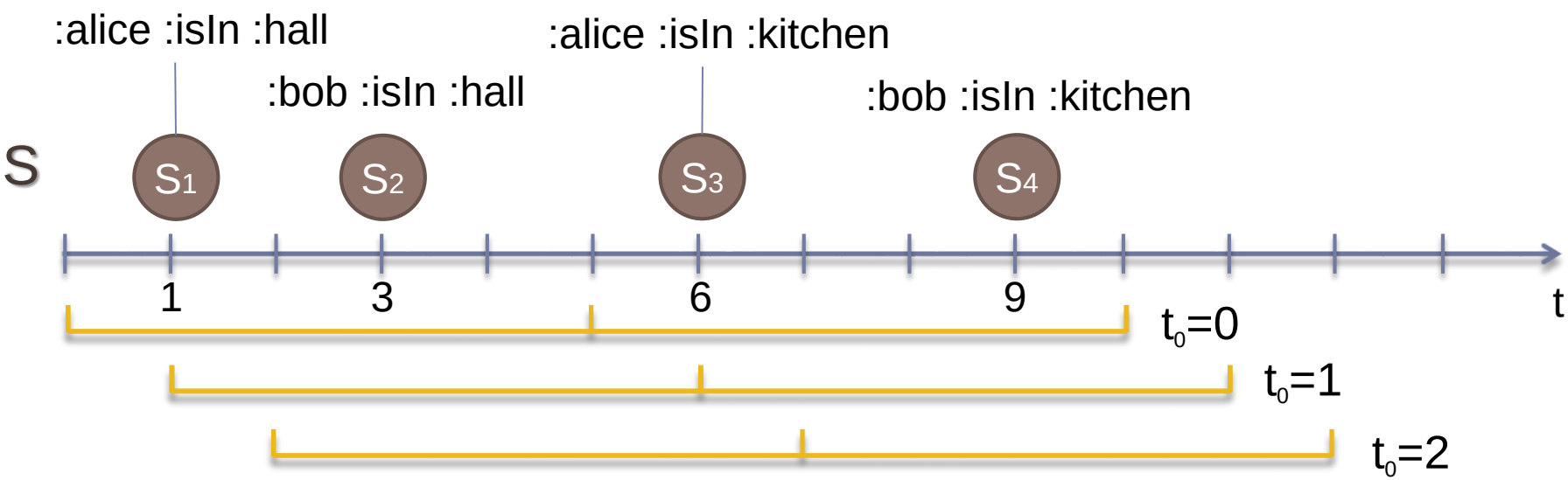
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]
t_0=1	:hall [6]	:kitchen [11]

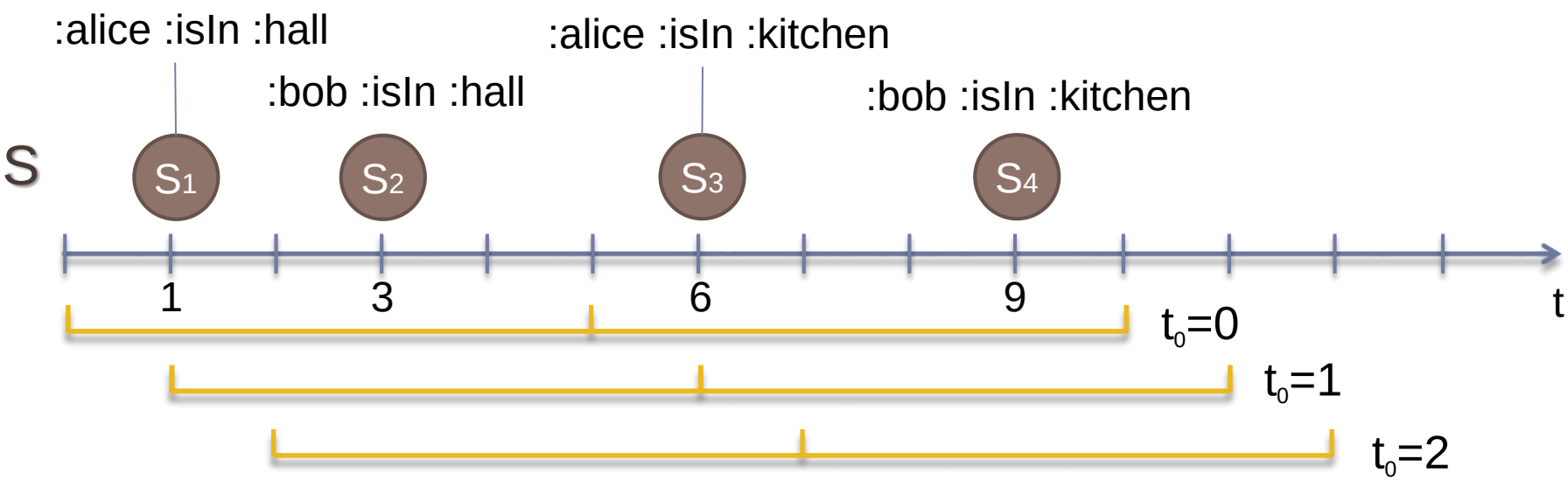
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]
t_0=1	:hall [6]	:kitchen [11]
t_0=2	- [7]	- [12]

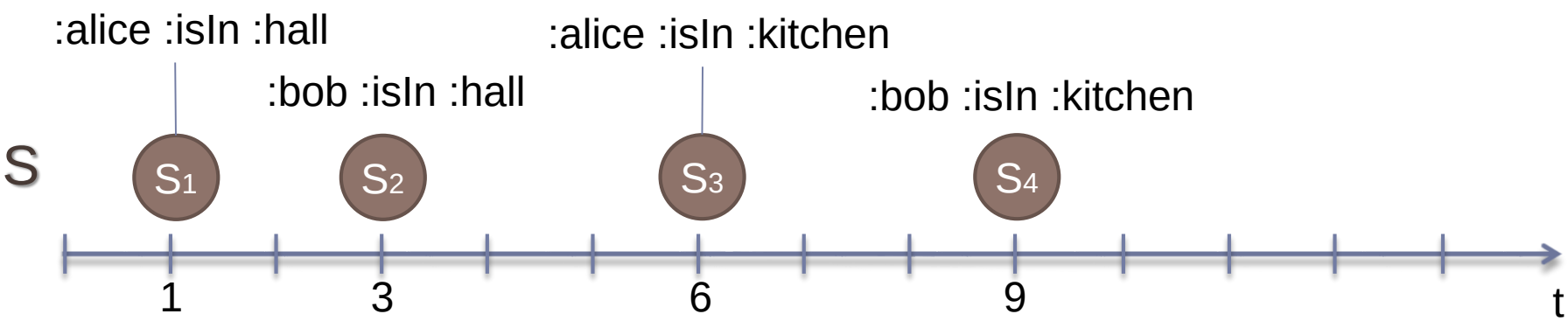
RSP output correctness



Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

Window	1° answer	2° answer
t_0=0	:hall [5]	:kitchen [10]
t_0=1	:hall [6]	:kitchen [11]
t_0=2	- [7]	- [12]

RSP output correctness



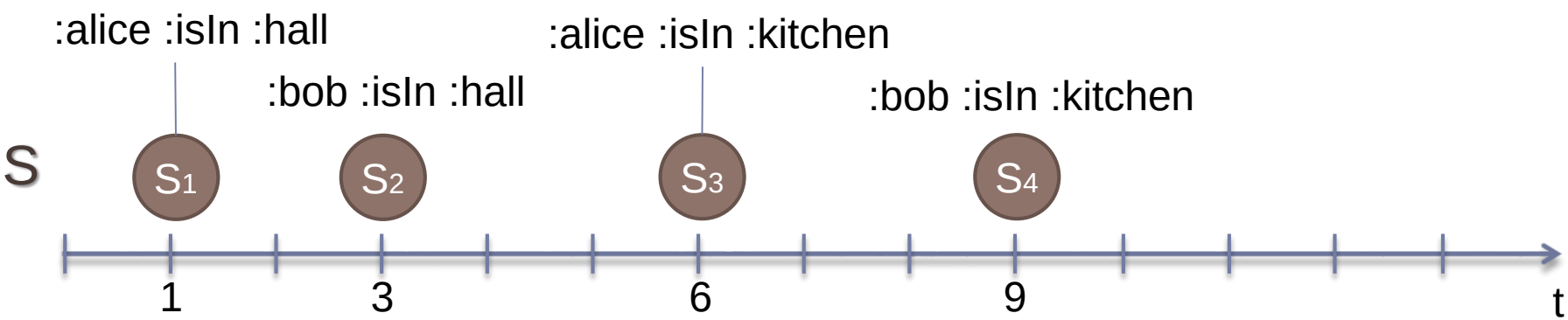
System 1

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

RSP output correctness



System 1

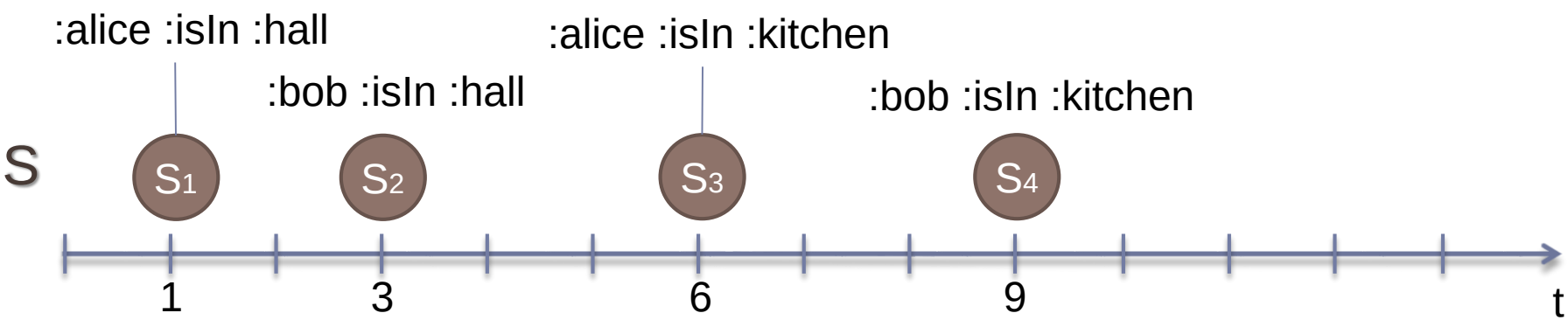
Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

Window-close vs content-change report policy

RSP output correctness



System 1

Execution	1° answer	2° answer
1	:hall [6]	:kitchen [11]
2	:hall [5]	:kitchen [10]
3	:hall [6]	:kitchen [11]
4	- [7]	- [12]

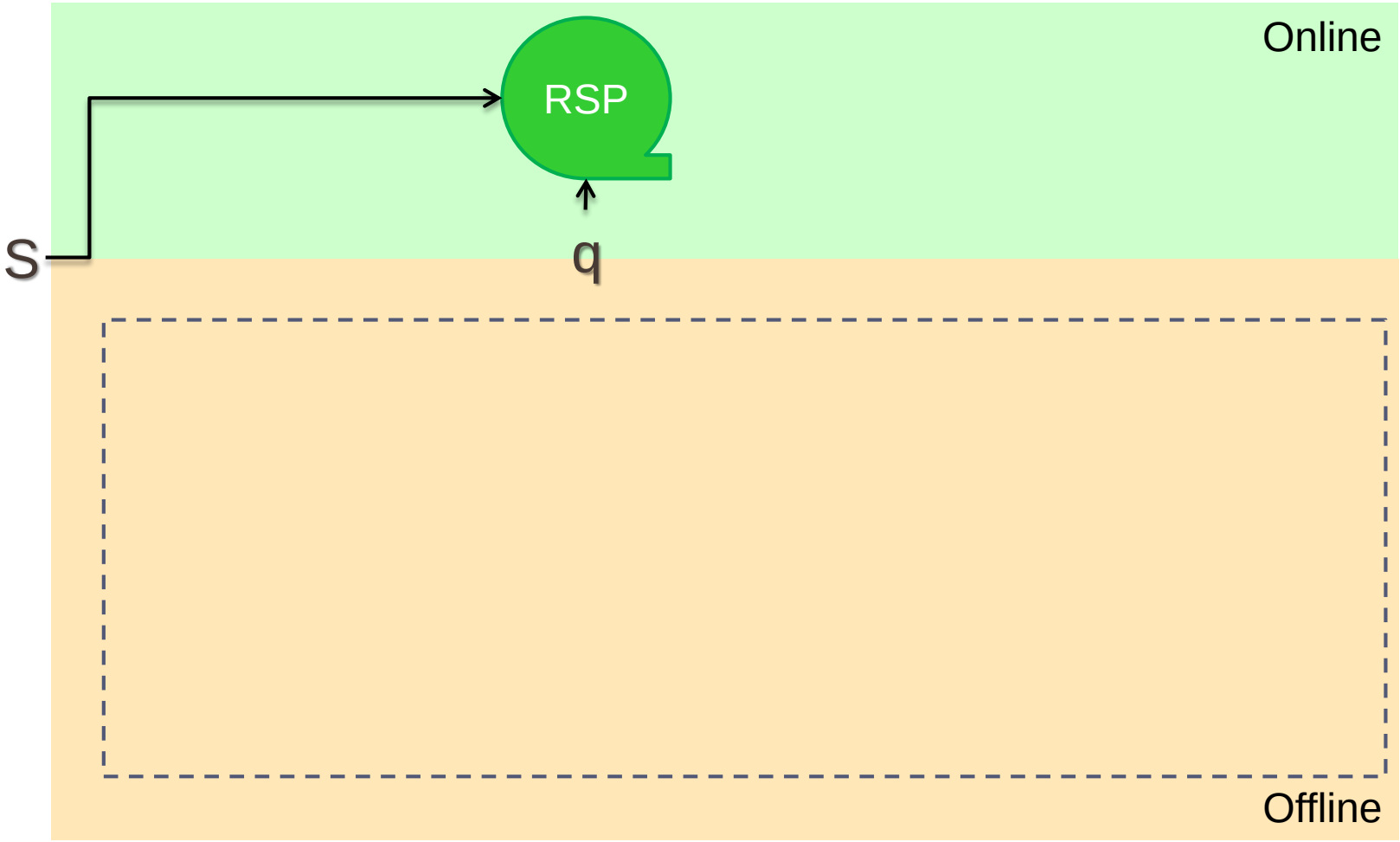
System 2

Execution	1° answer	2° answer
1	:hall [3]	:kitchen [9]
2	No answers	
3	:hall [3]	:kitchen [9]
4	No answers	

Window-close vs content-change report policy

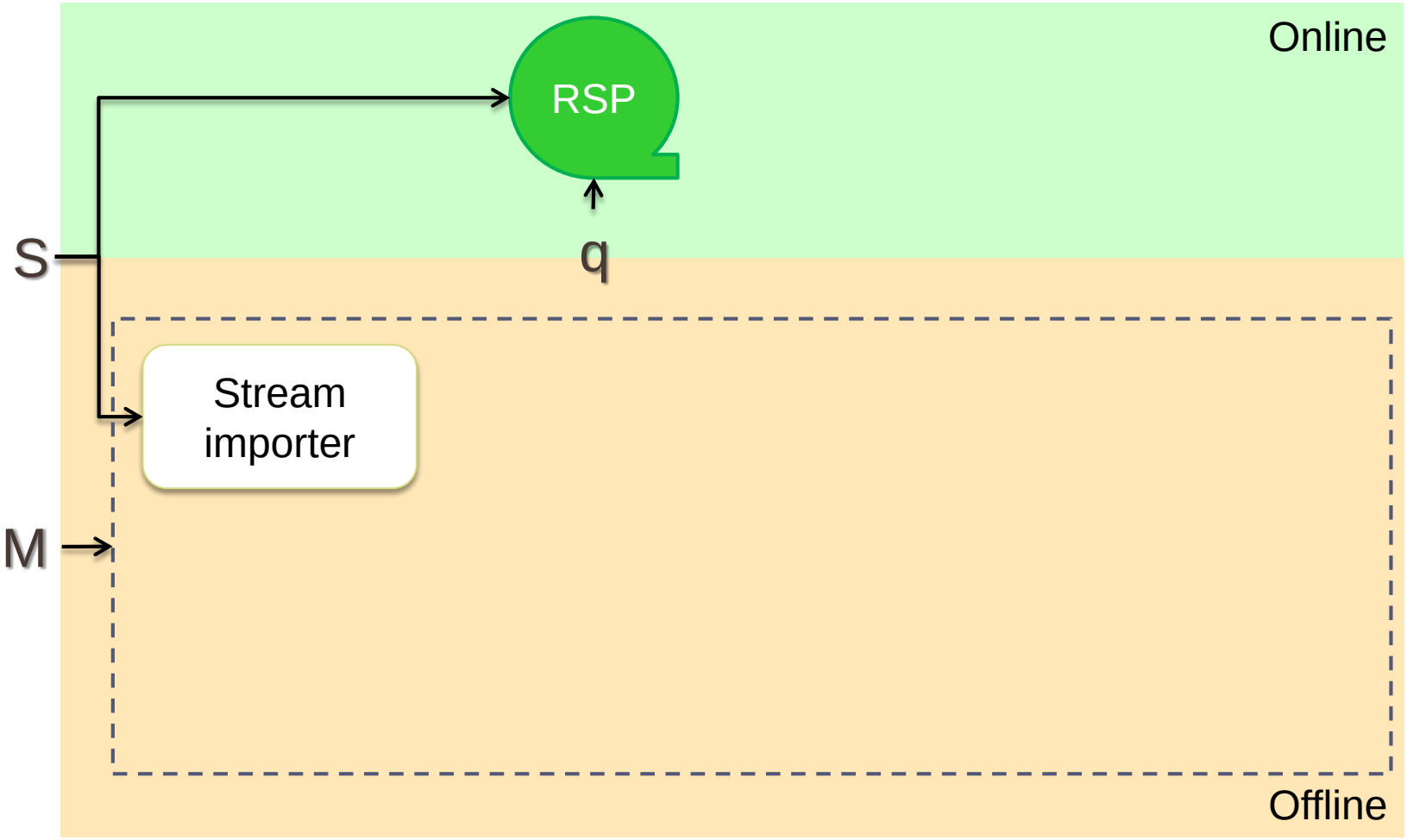
Empty relation notification (yes|no)

The oracle



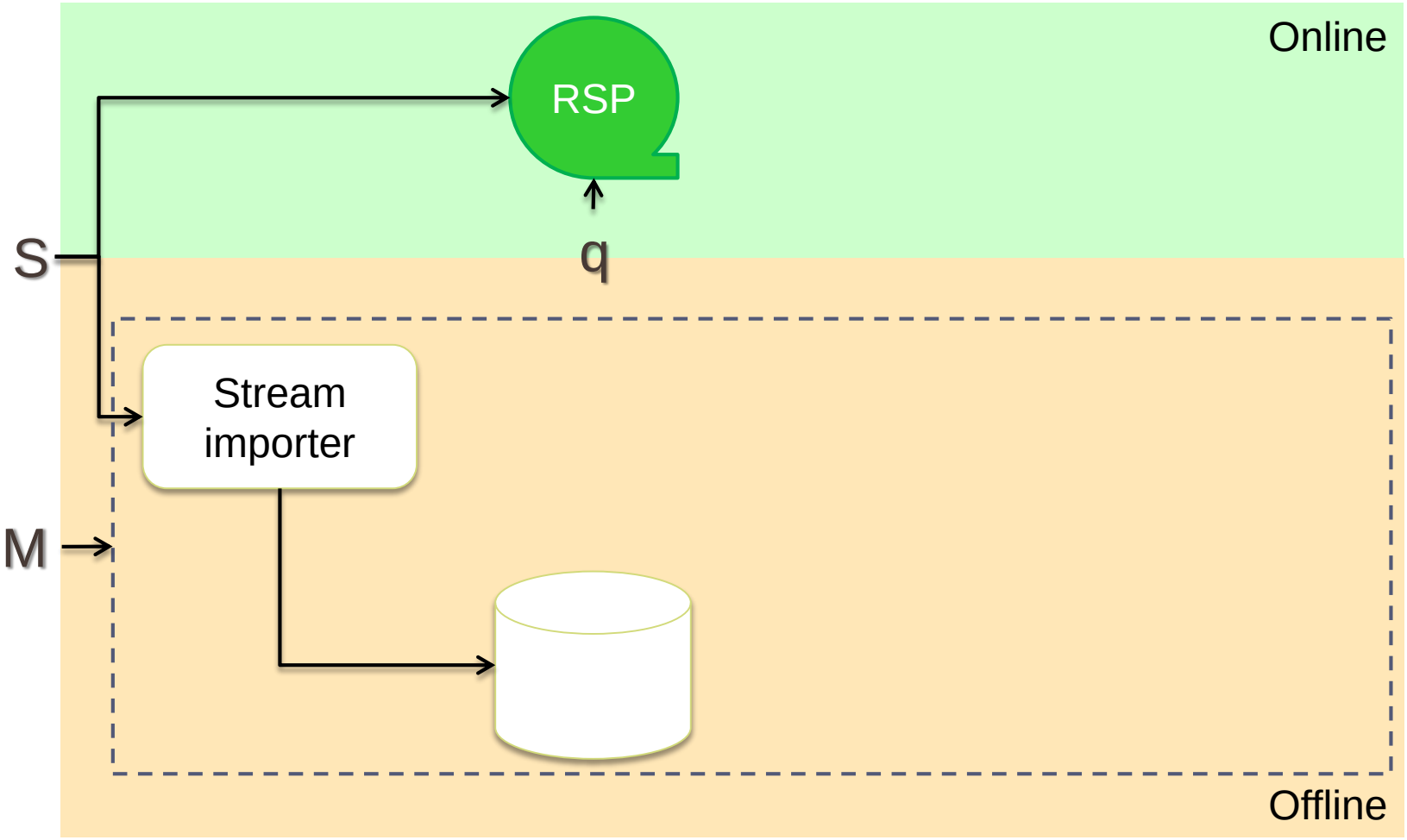
Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



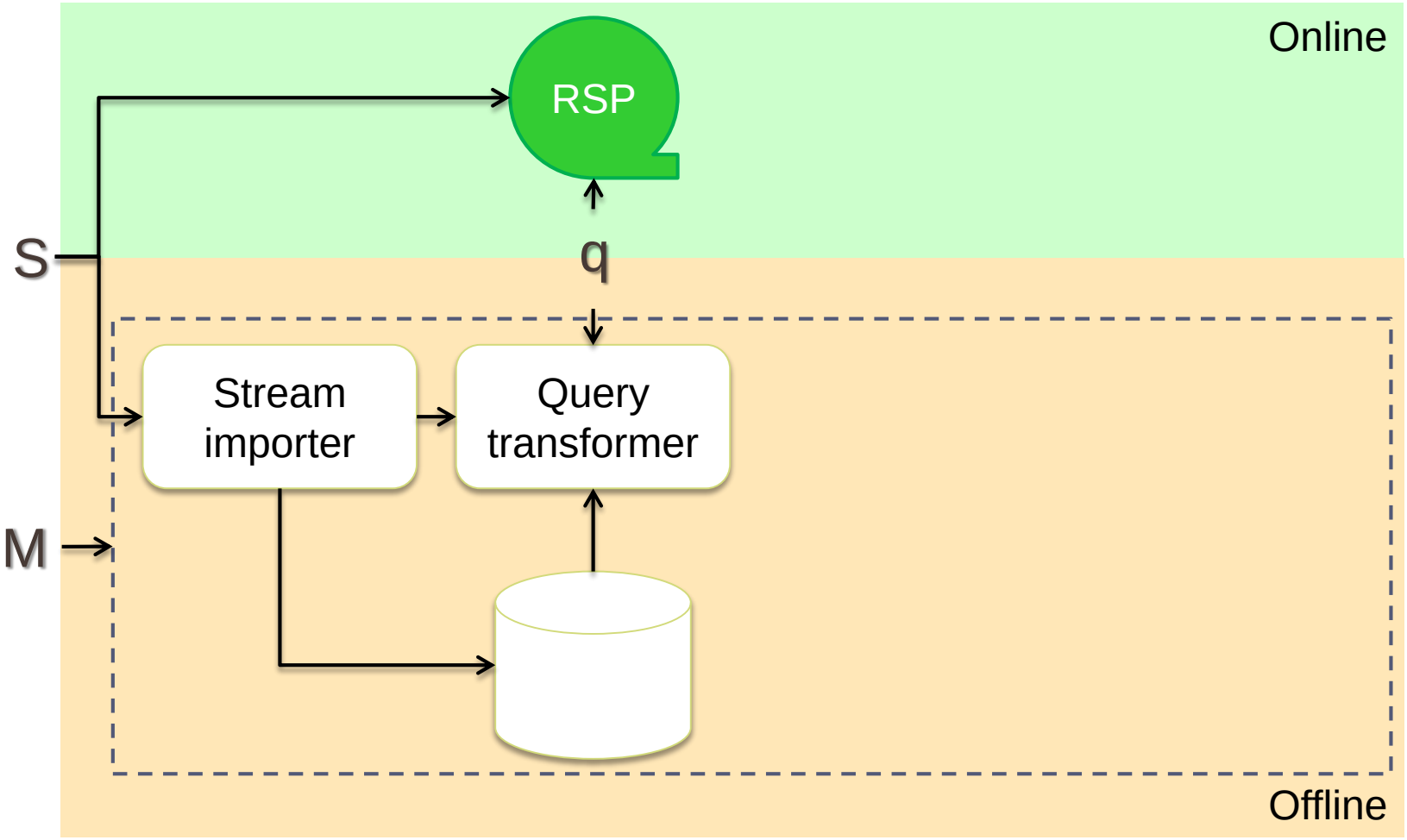
Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



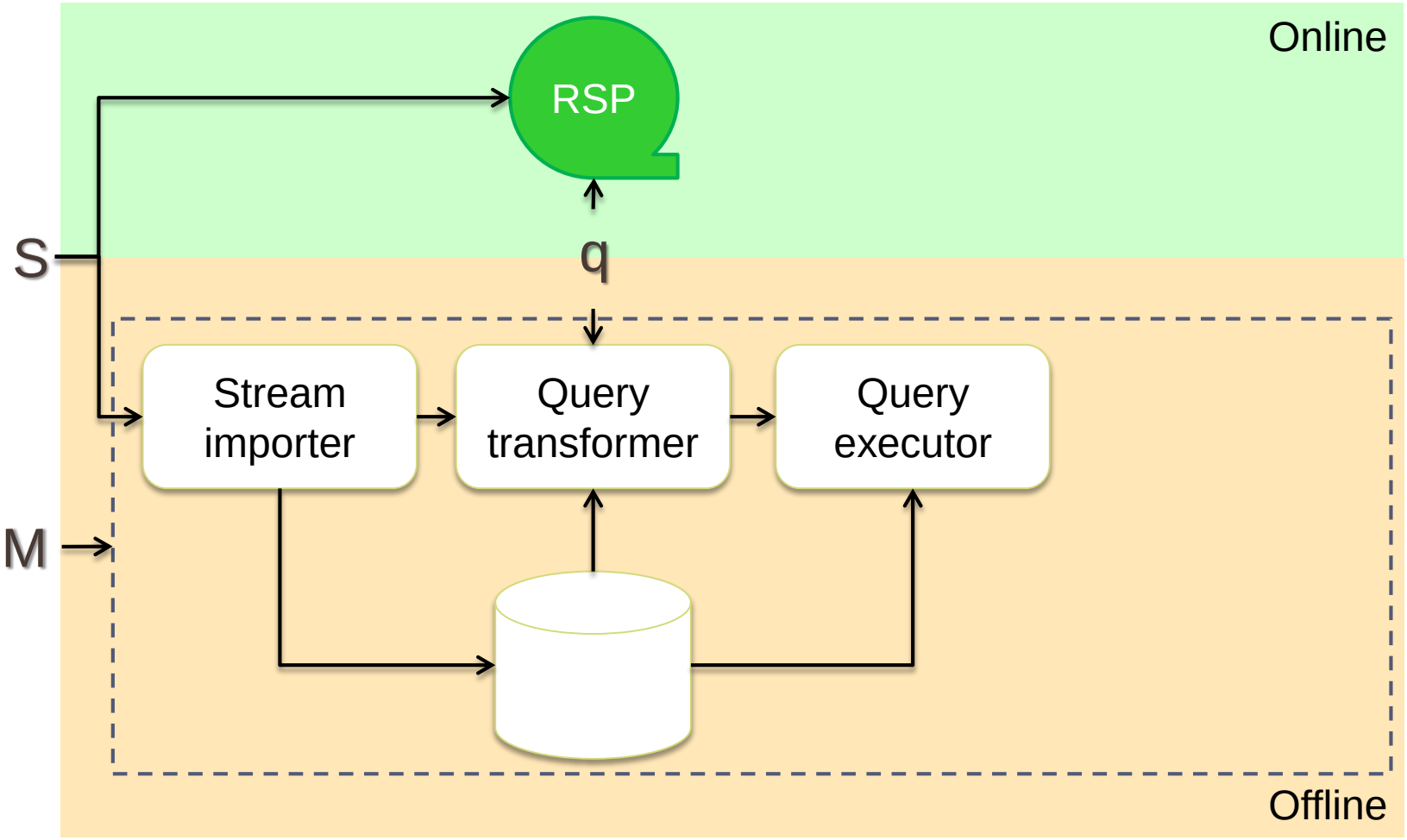
Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



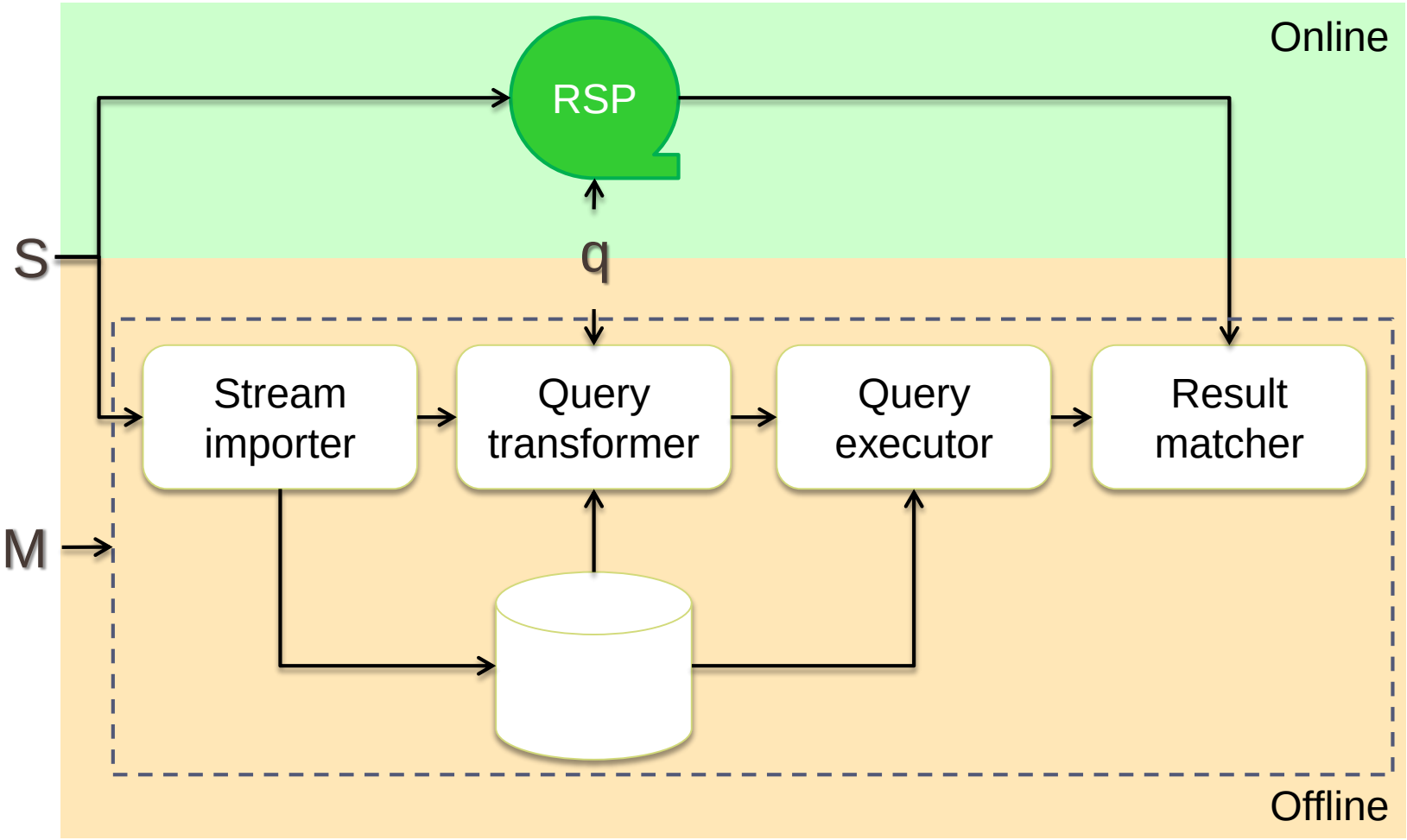
Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



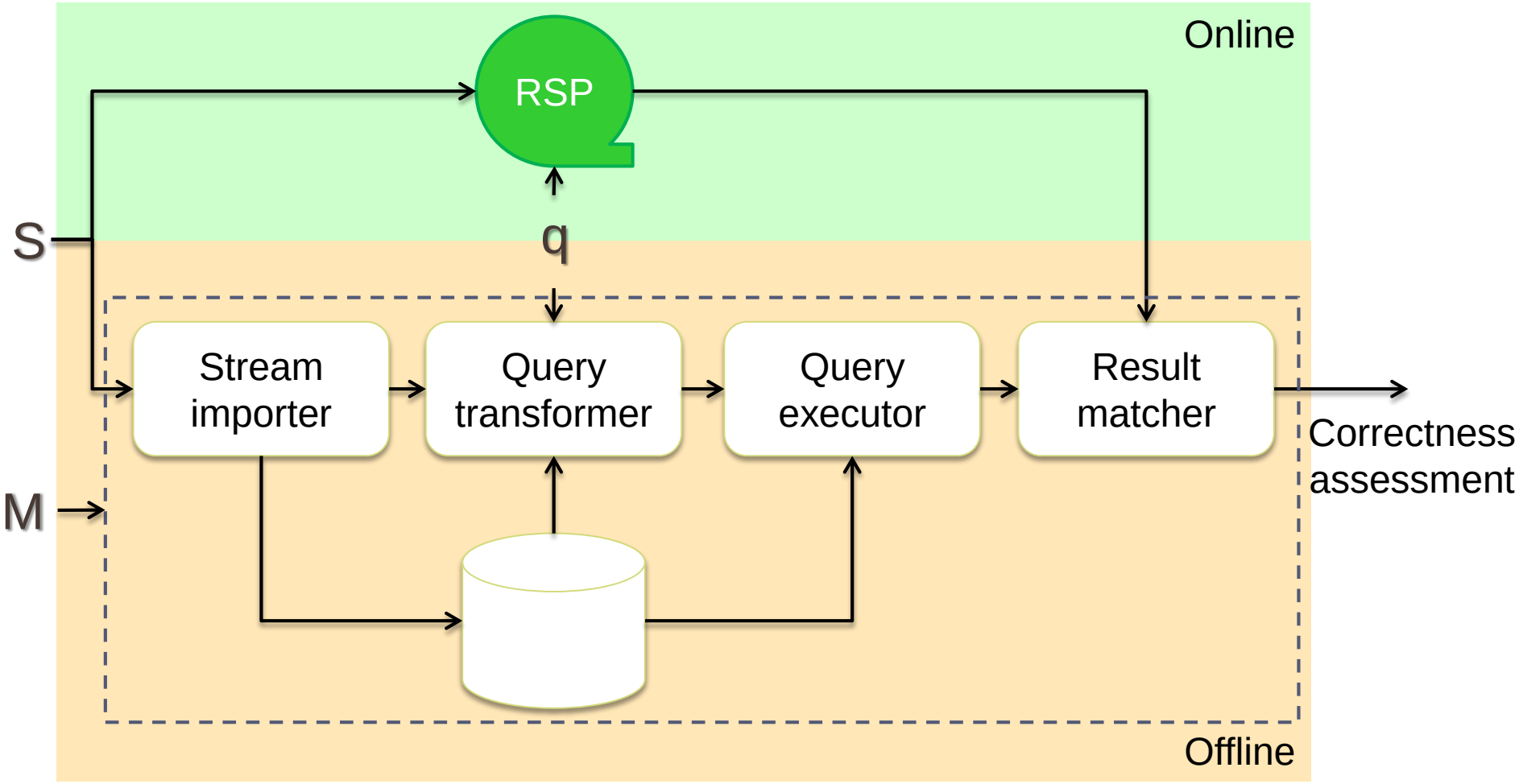
Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

The oracle



Available at: https://github.com/dellaglio/csrbench-ora
(Apache 2.0 licence)

Design of the tests

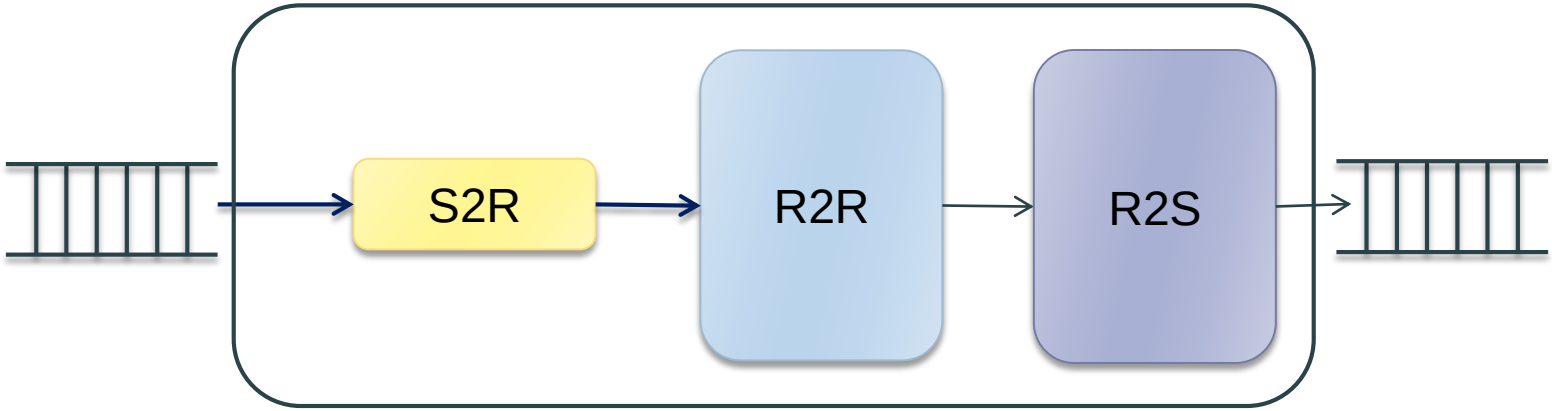
- As data set, we consider the LinkedSensorData data set
  - Data stream describing blizzards and hurricanes in the US
- We designed the query set taking into account
  - Window size and slide parameters
  - Presence of aggregation operators
  - Joins of timestamped triples
- We collected a set of seven parametrized queries
- The list of the queries and their explanation is available at: http://www.w3.org/wiki/CSRBench

Results

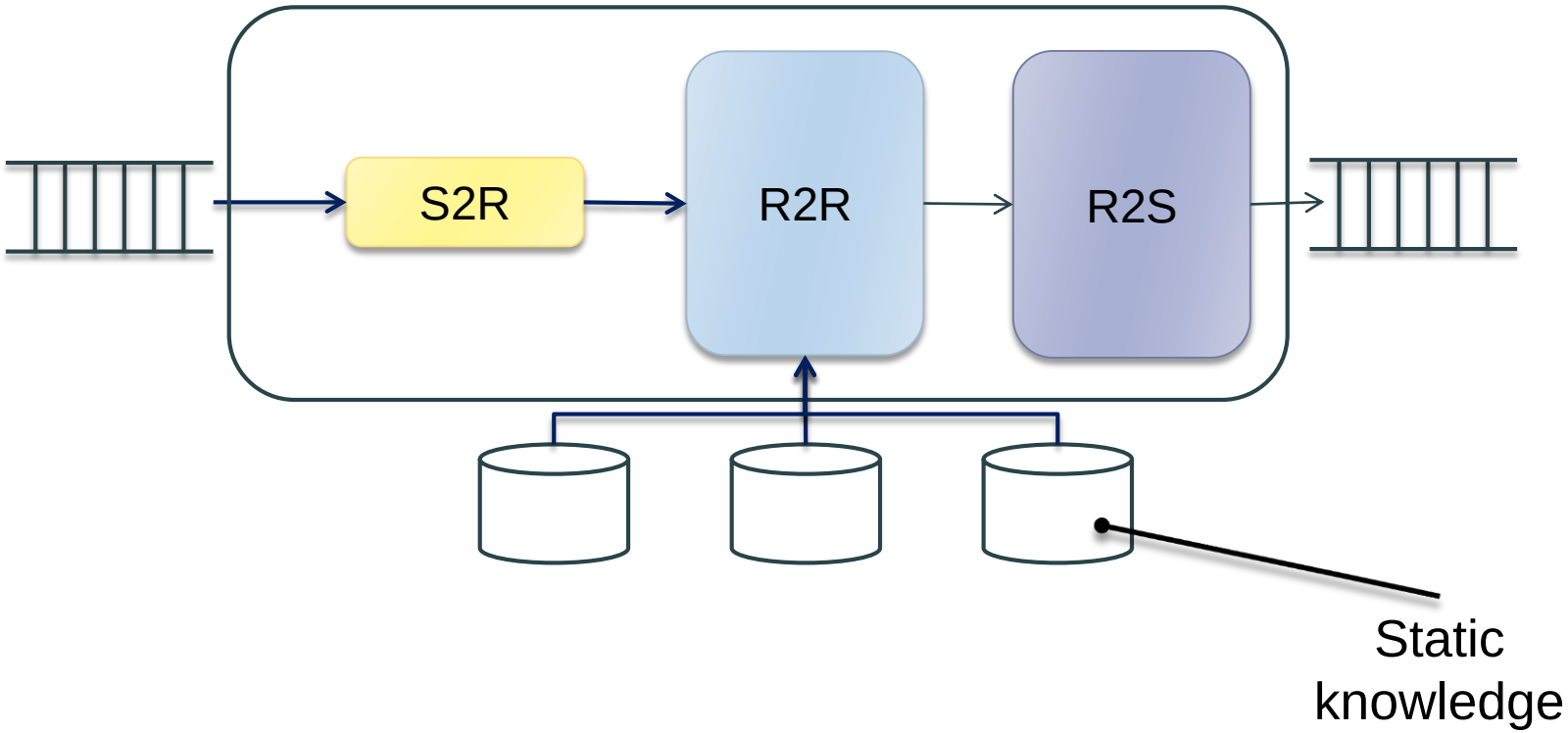
- All the three systems that we considered in our experiments showed wrong behaviours
- The defects we identified are related to:
  - the window operator
    - Initialization
    - Slide parameter
    - Window contents
  - timestamps of the triples
    - Internal timestamp management

	CQELS	C-SPARQL	SPARQL$_{stream}$
Q1			
Q2			
Q3			
Q4			
Q5			
Q6			
Q7			

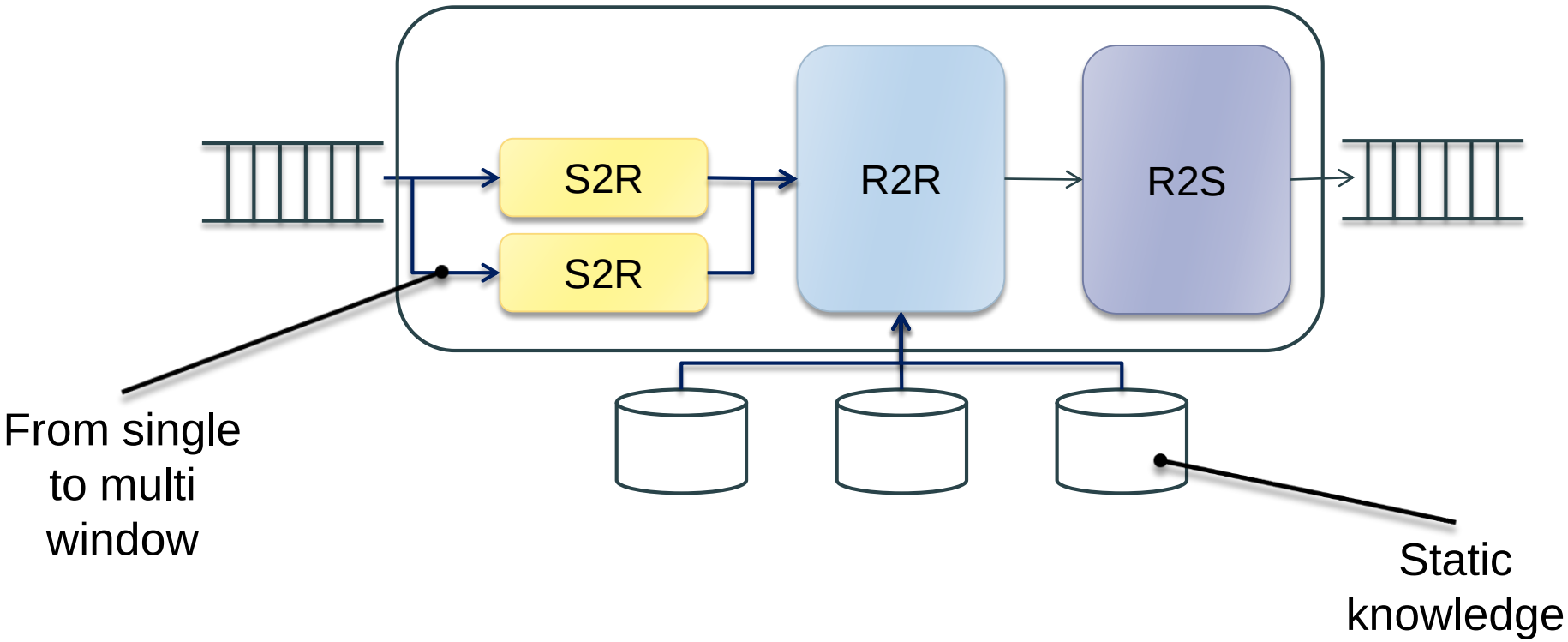
(Removable) constraints



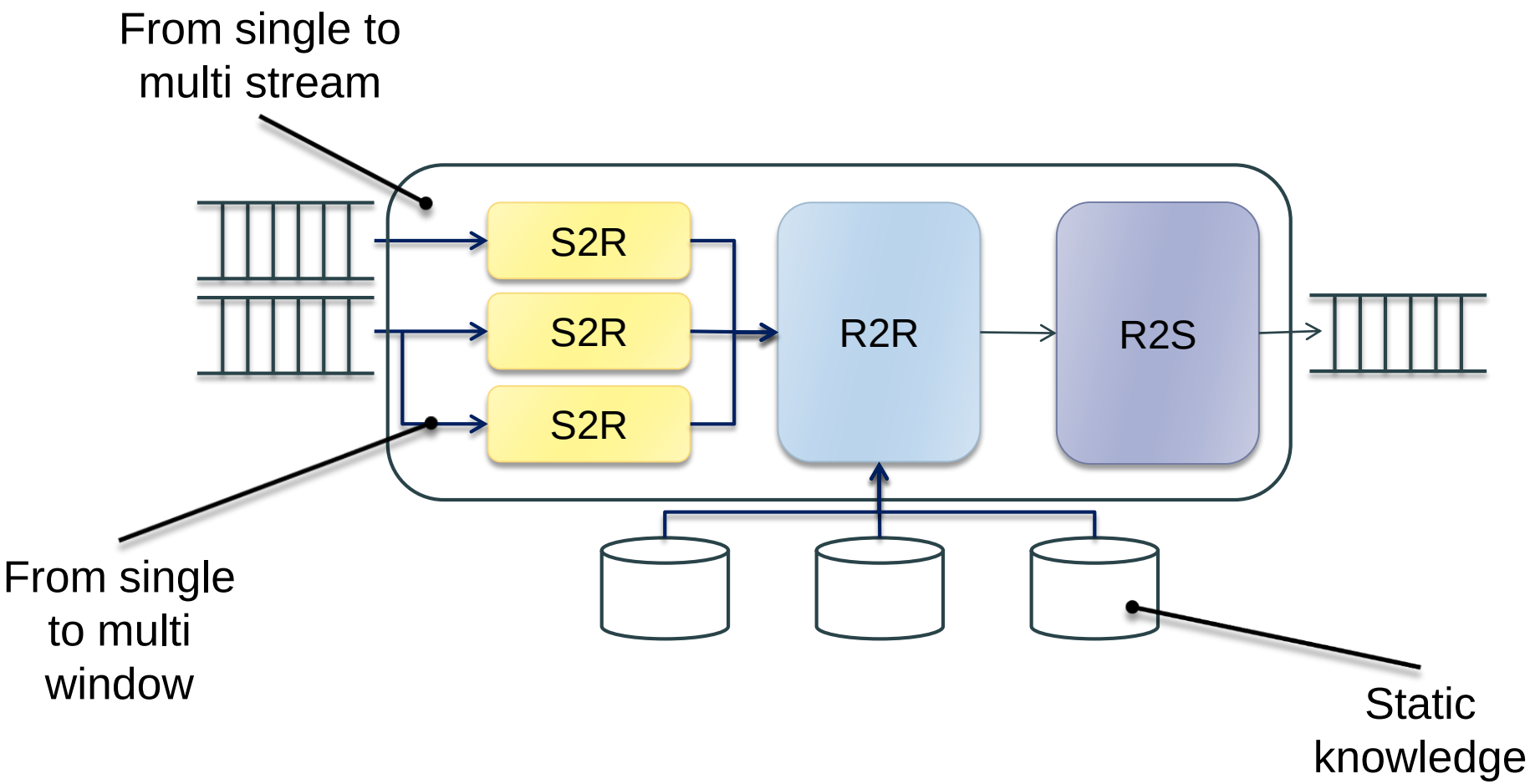
(Removable) constraints



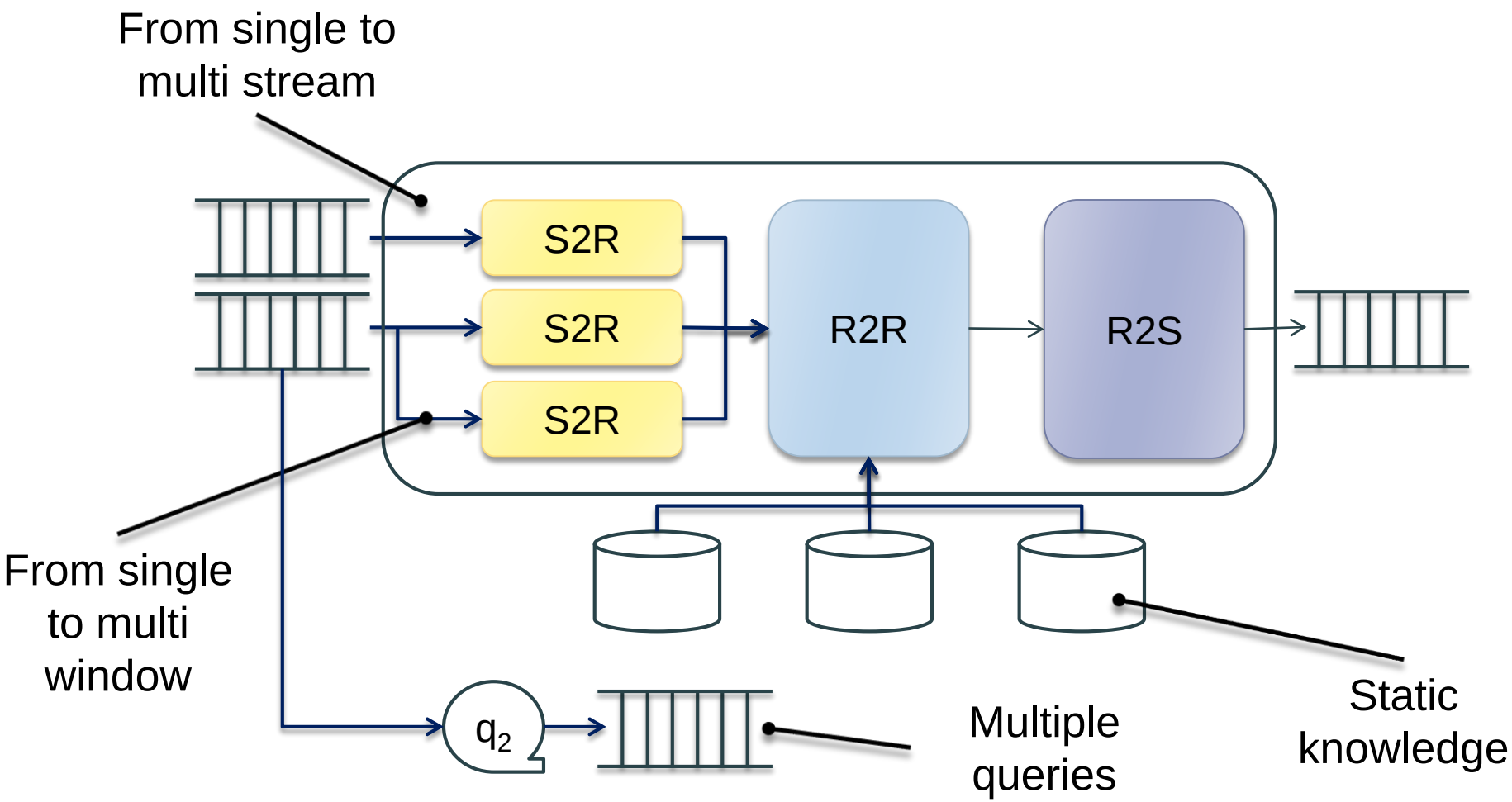
(Removable) constraints



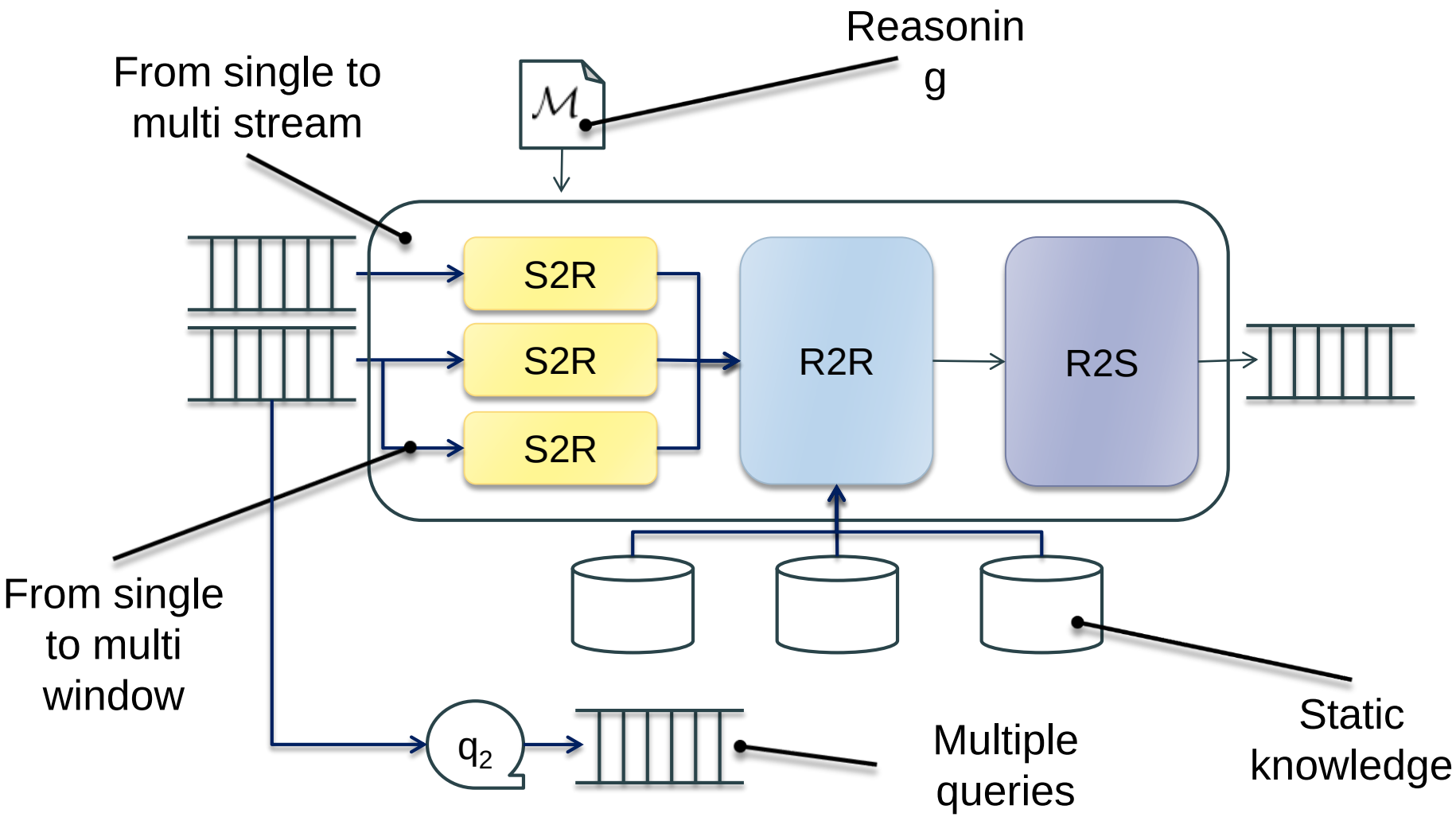
(Removable) constraints



(Removable) constraints



(Removable) constraints



Conclusions

- A **model** that describes in a **more accurate** way the RSPs' **operational semantics** helps their improvement:
  - Better design of the system
  - Prediction of the expected behaviours
- A **common and shared test environment** helps developers of both existing and upcoming RSPs
  - It becomes easier to set up experiments to detect defects and to correct them
- Possible **improvements** of the **existing benchmarks** (e.g., SRbench and LSBench)
  - Design of new tests
  - Better interpretation of the experiment results

Thank you! Questions?

On correctness in RDF stream processor benchmarking

Daniele Dell'Aglio (DEIB, Politecnico di Milano)

Jean-Paul Cabilmonte (OEG, Universidad Politécnica de Madrid)

Marco Balduini (DEIB, Politecnico di Milano)

Oscar Corcho (OEG, Universidad Politécnica de Madrid)

Emanuele Della Valle (DEIB, Politecnico di Milano)

wiki: http://www.w3.org/wiki/CSRBench

software: https://github.com/dellaglio/csrbench-oracle