



# Web Stream Processing with RSP4J

The Web Conf 2022

Pieter Bonte (BE) & Matteo Belcao (IT) & Marco Balduini (IT) & Emanuele Della Valle (IT)

26/04/2022

# Agenda

## Processing

- Stream Processing 101
  - Operator Families
  - Execution Semantics
- RSP-QL
- RSP4J, YASPER, & CO
- RSP Internals
- Demo and Exercises



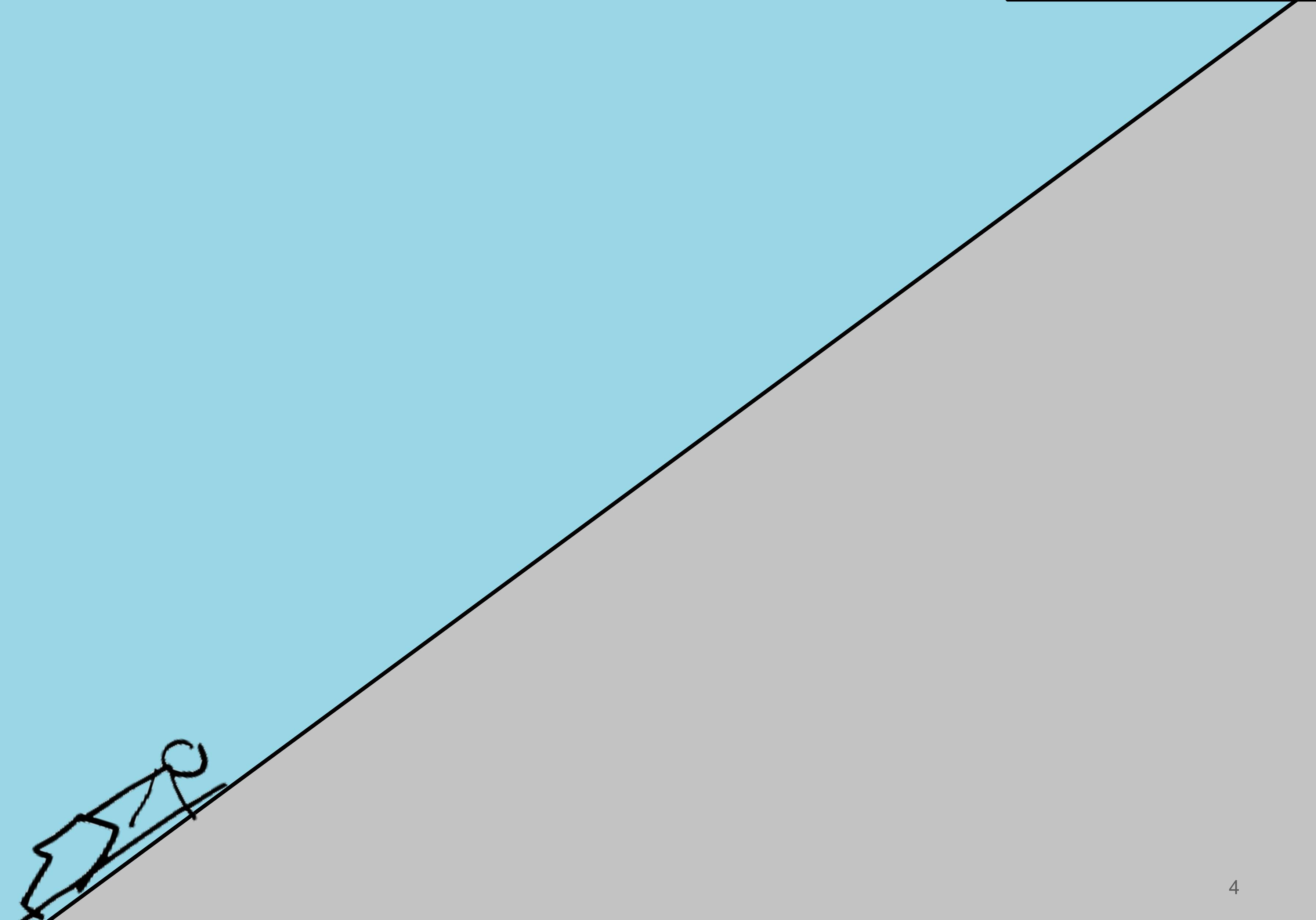
There will be exercises!

# You want to follow the exercises?

- Requirements:
  - Maven
  - Java 9+
- Clone the git repo: <https://github.com/pbonete/WSP-TheWebConf2022Tutorial>
- Open with your favourite Java IDE (e.g. Eclipse/IntelliJ)



# Stream Processing

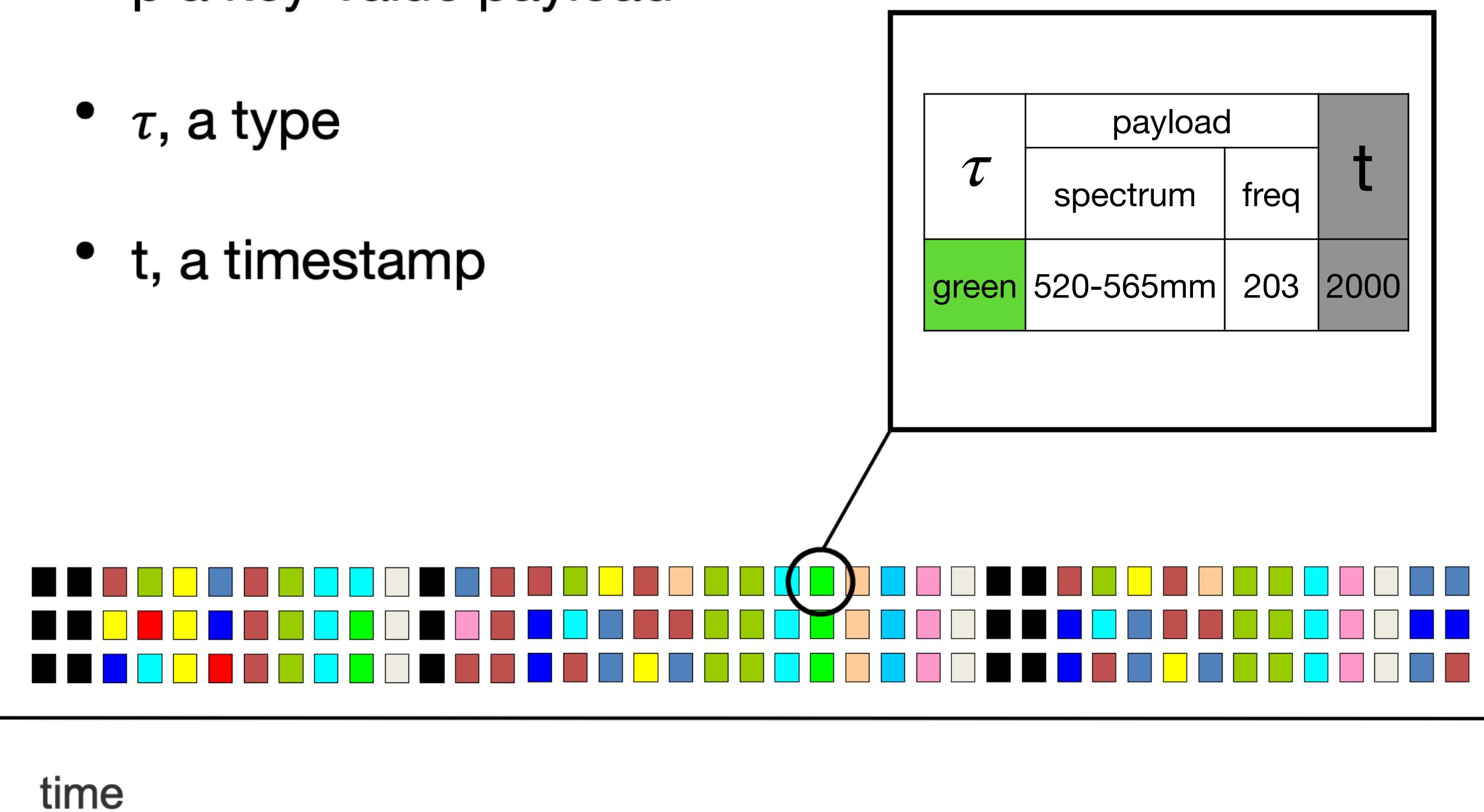


# CQL Abstractions

## Stream

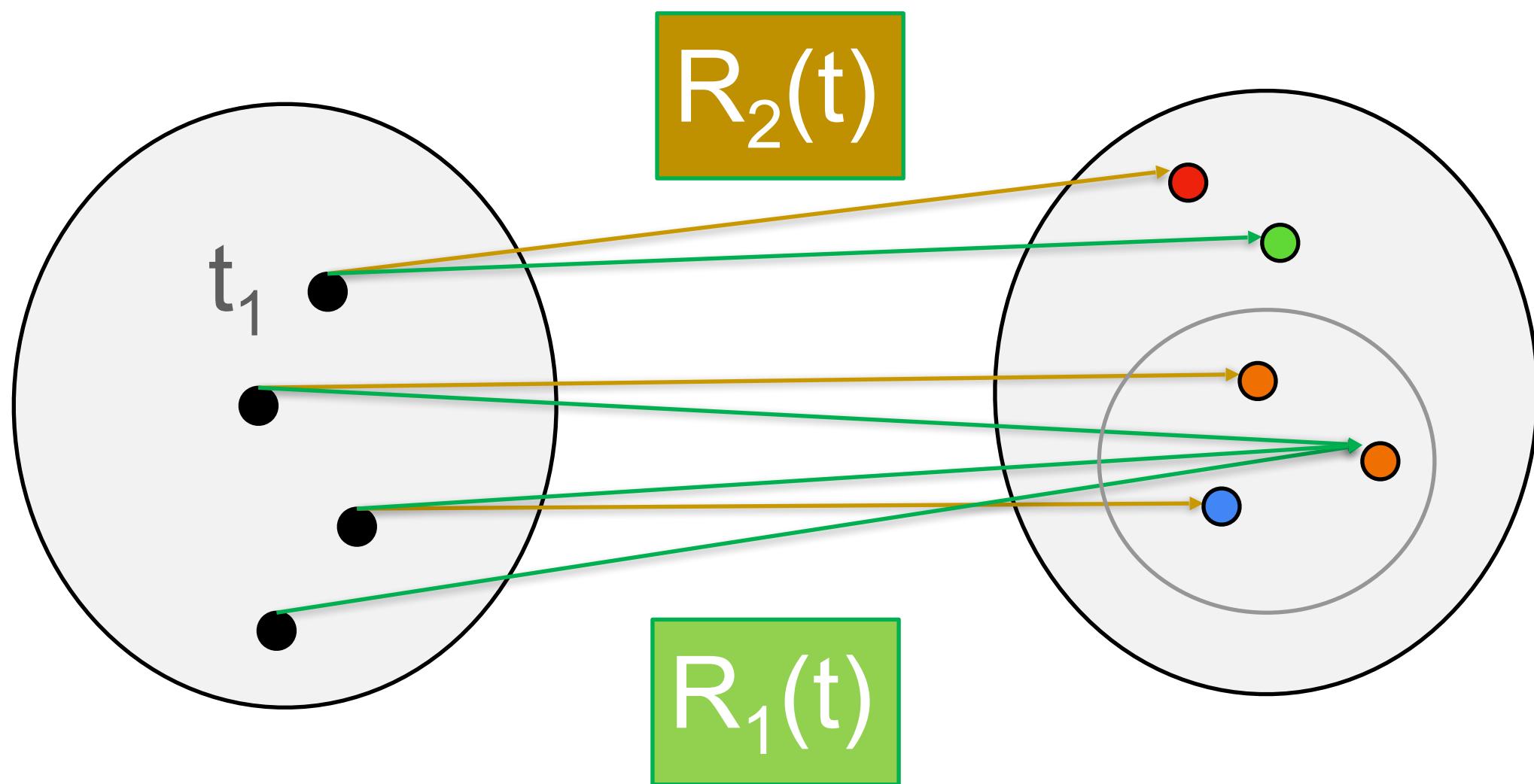
- A Stream S is a possibly infinite multi-set of elements  $\langle s, t \rangle$  where
  - $s$  is a tuple belonging to the schema of S and
  - $t$  is a timestamp.

- $p$  a key-value payload
- $\tau$ , a type
- $t$ , a timestamp



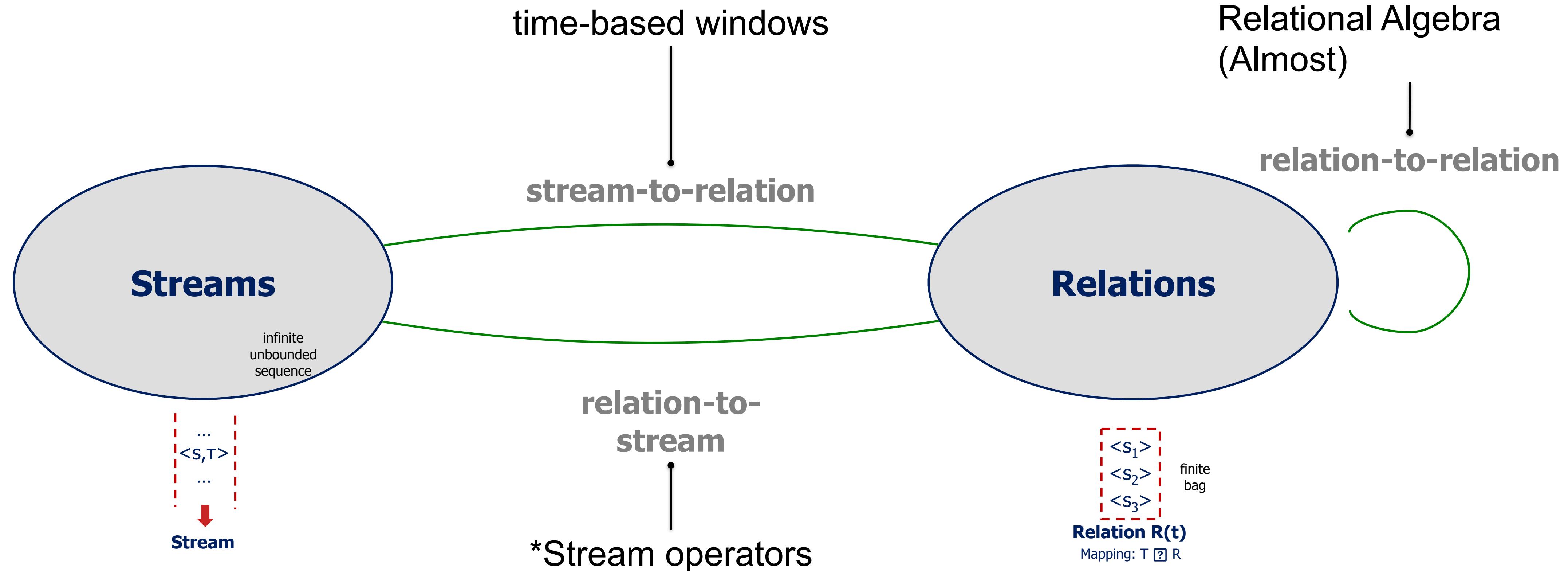
# CQL Abstractions

## Time-Varying Relation



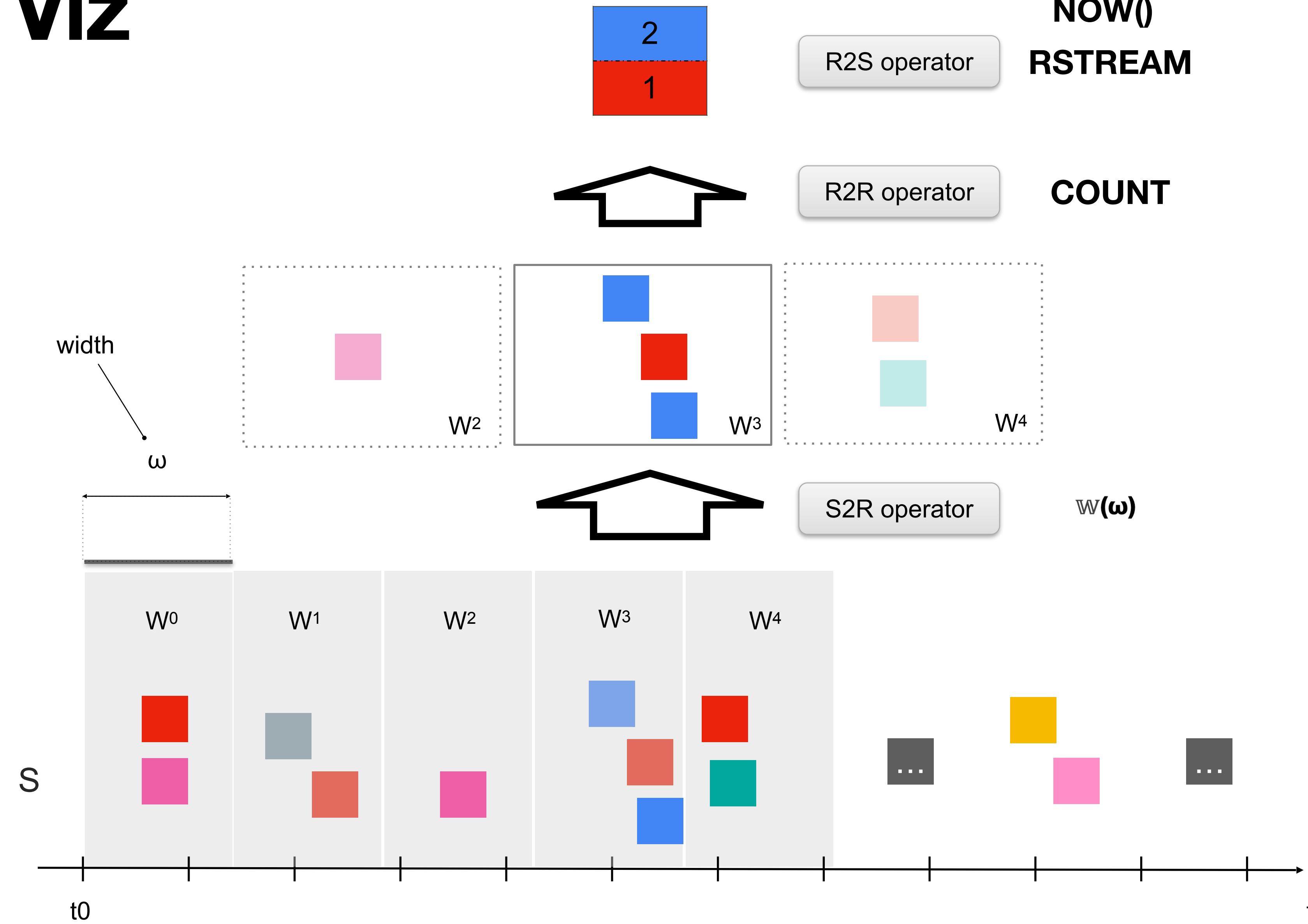
Time-Varying Relation  $R$  is a mapping from each time instant in  $T$  to a finite but unbounded bag of tuples belonging to the schema of  $R$ .

# The CQL: Families of Operators



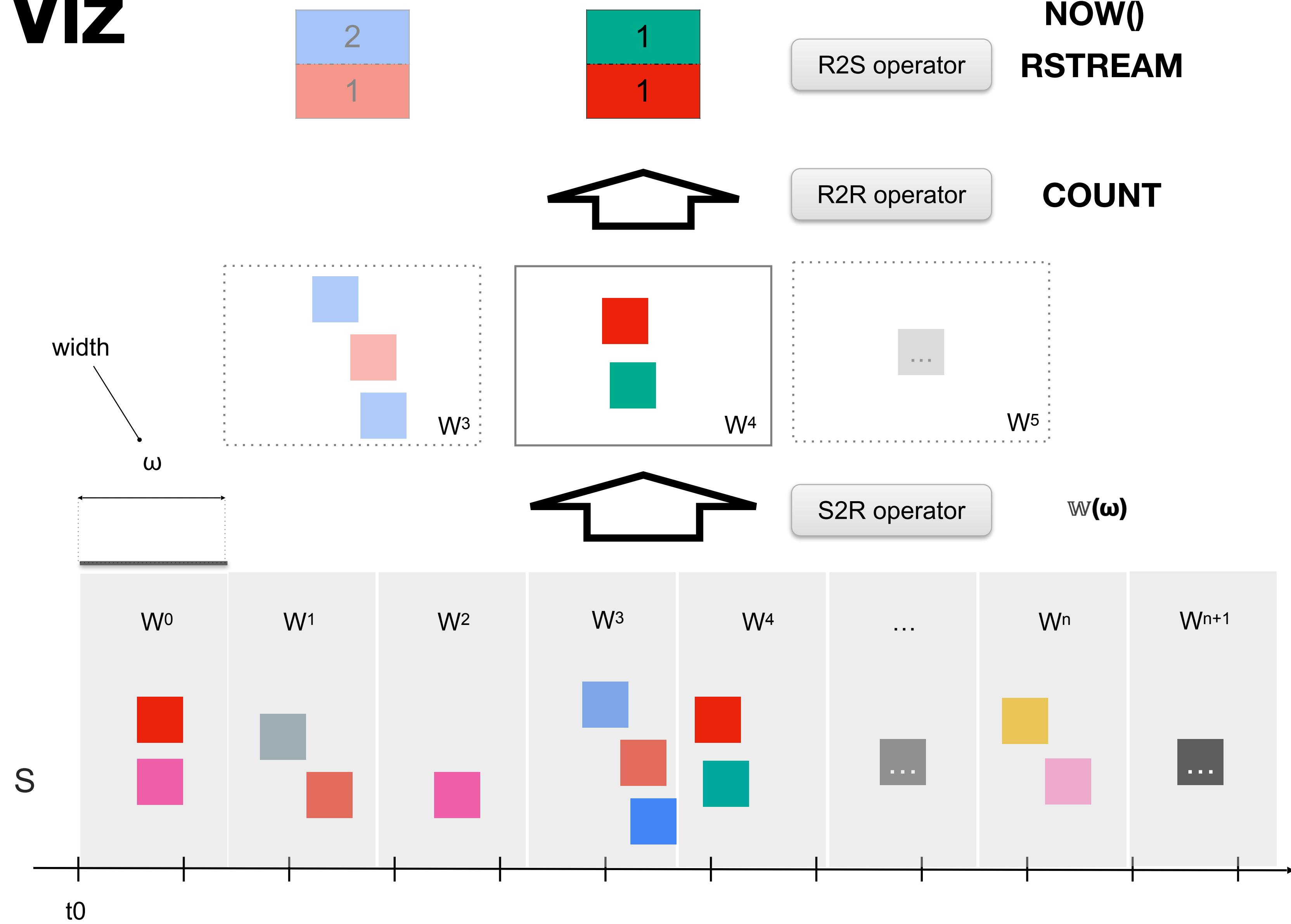
# CQL Viz

## S2R



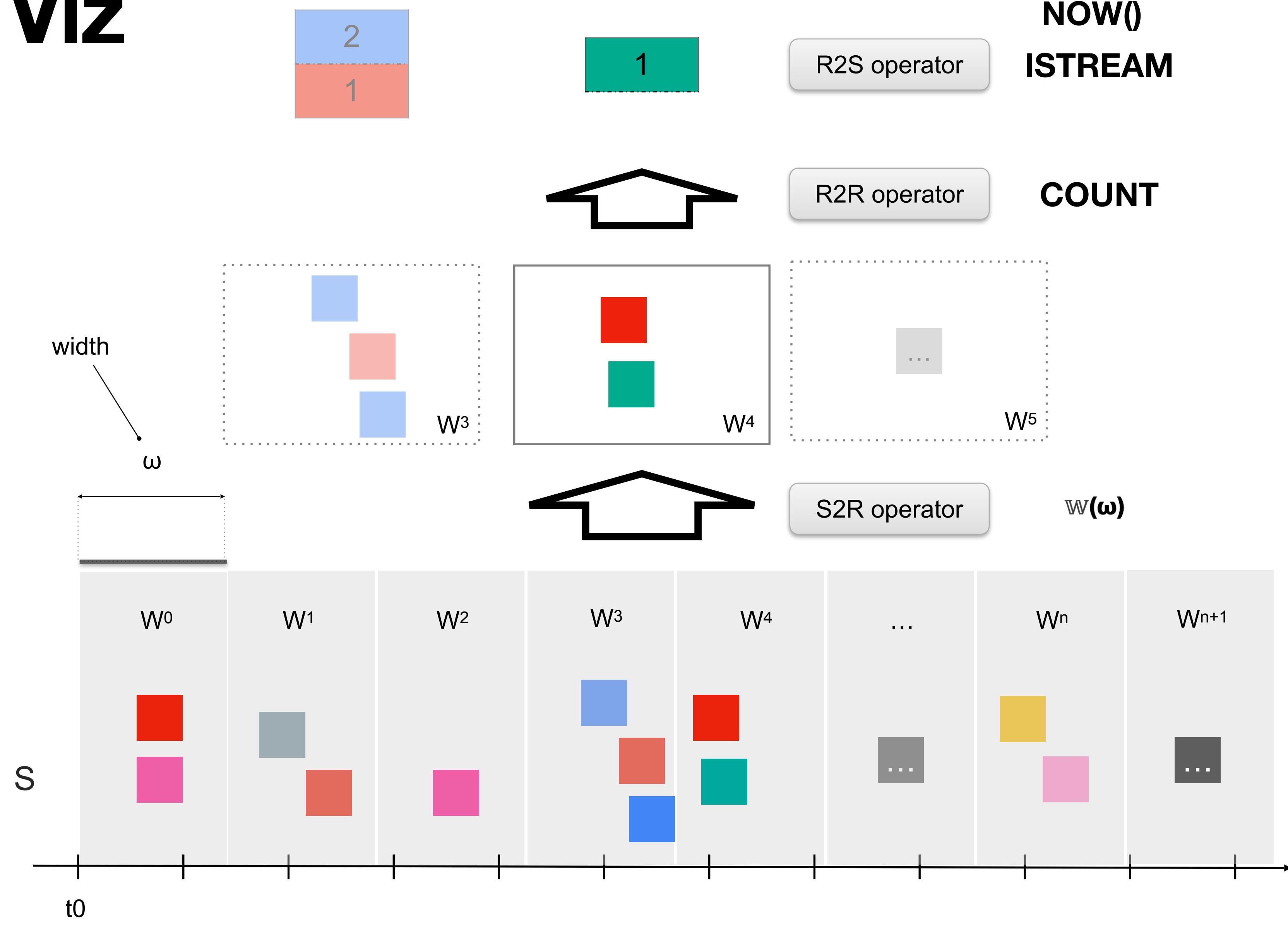
# CQL Viz

## S2R



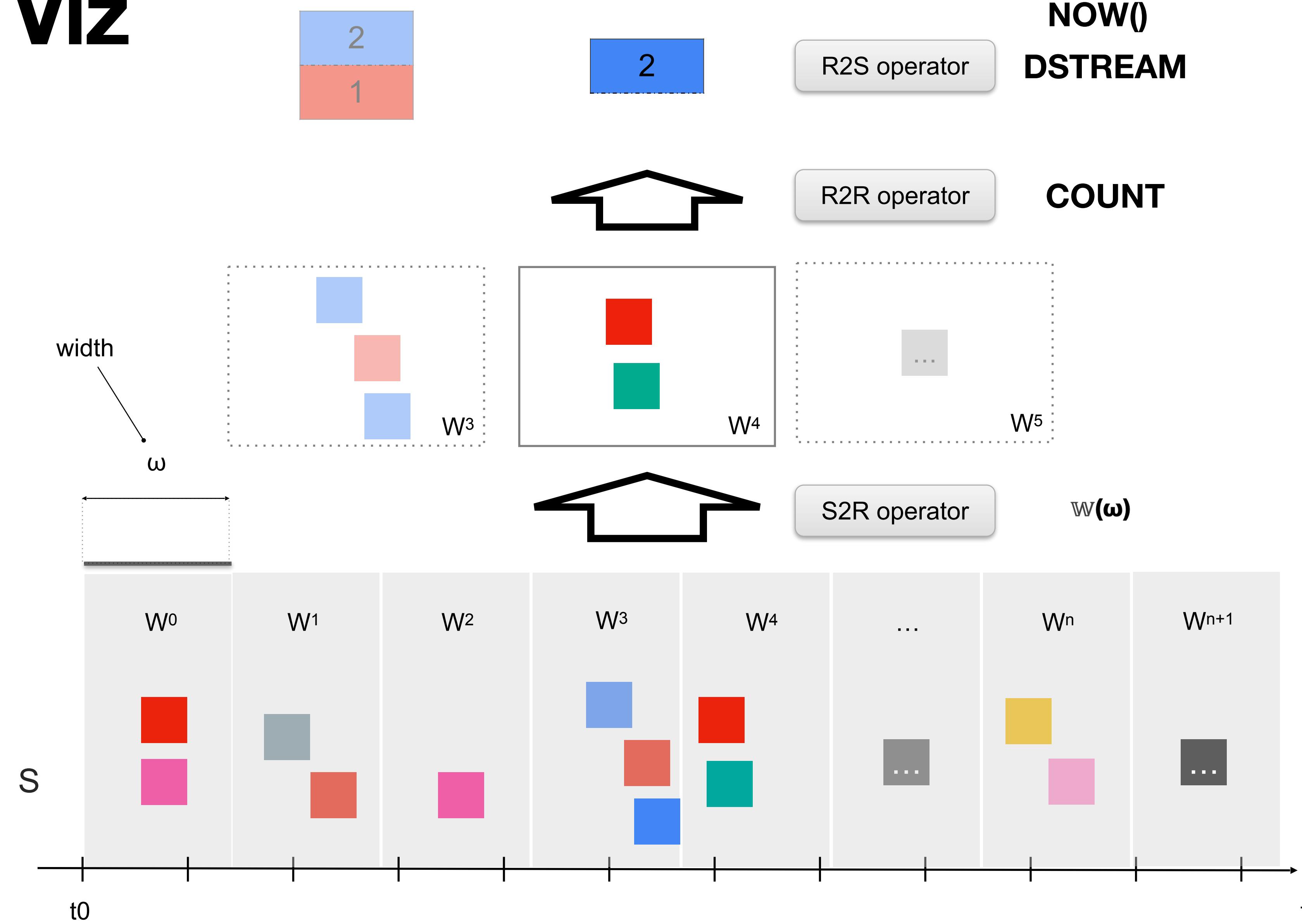
# CQL Viz

## R2S



# CQL Viz

## S2R

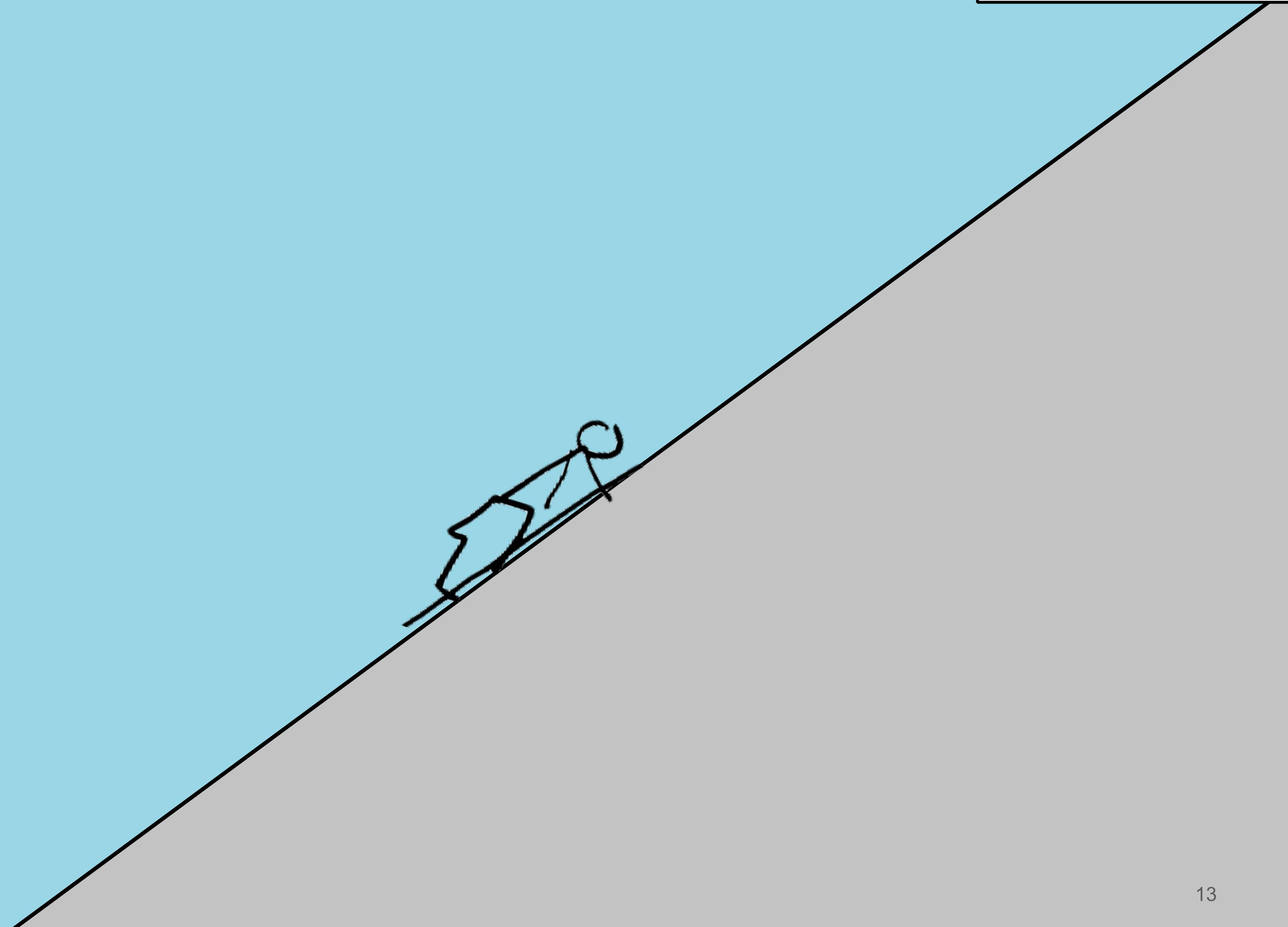


Continuous●Semantics

/kən'tɪnju:əs●sɪ'mæntɪks/

*1. The result of a query is the set of results that would be returned if the query were executed at every instant in time.*

# Stream Processing



# Execution Semantics

## The Data Flow Model

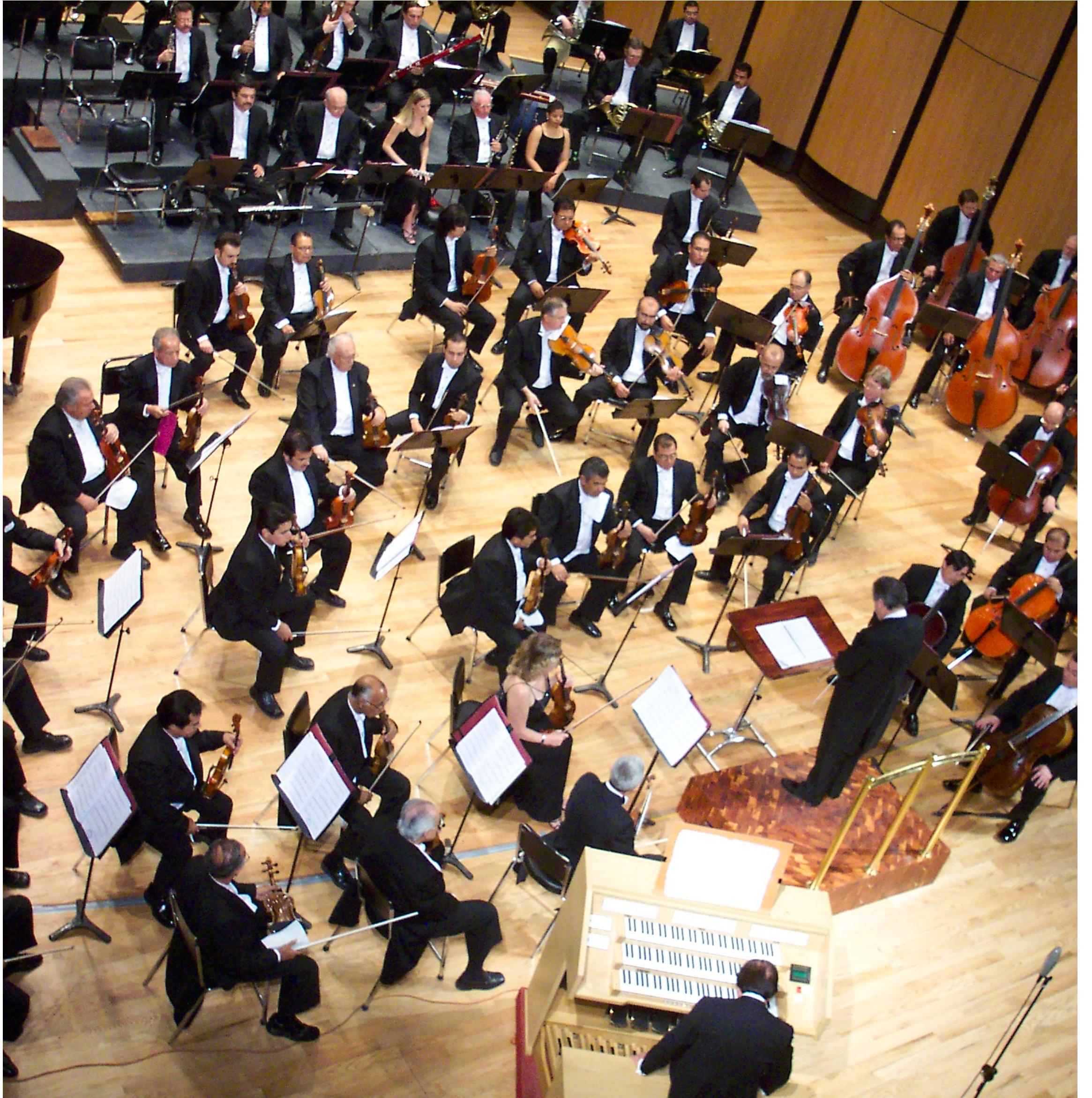
- Describe how the data system behave over time
  - **What** results are being computed
  - **Where** in event time are being computed
  - **When** in processing time are being materialised
  - **How** “earlier results” relate to “later refinements”



# Execution Semantics

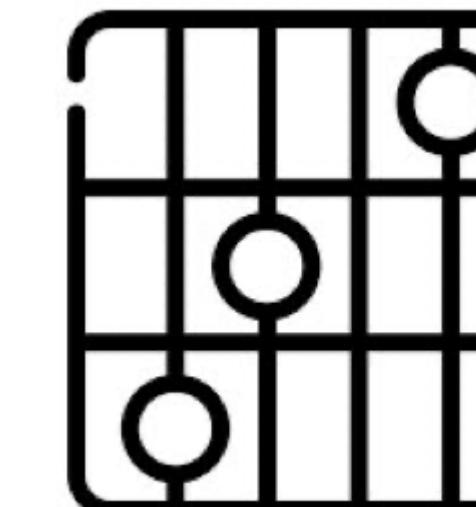
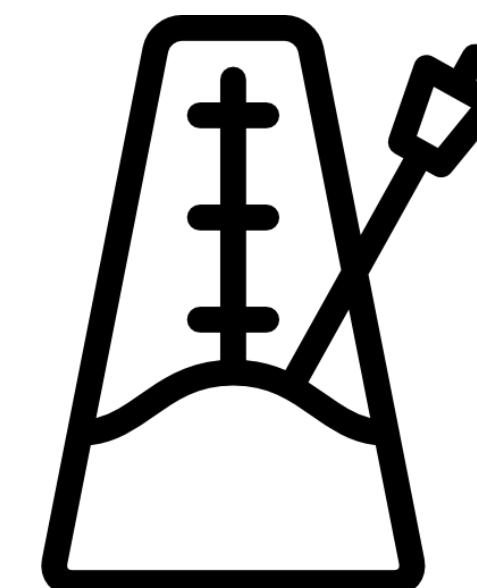
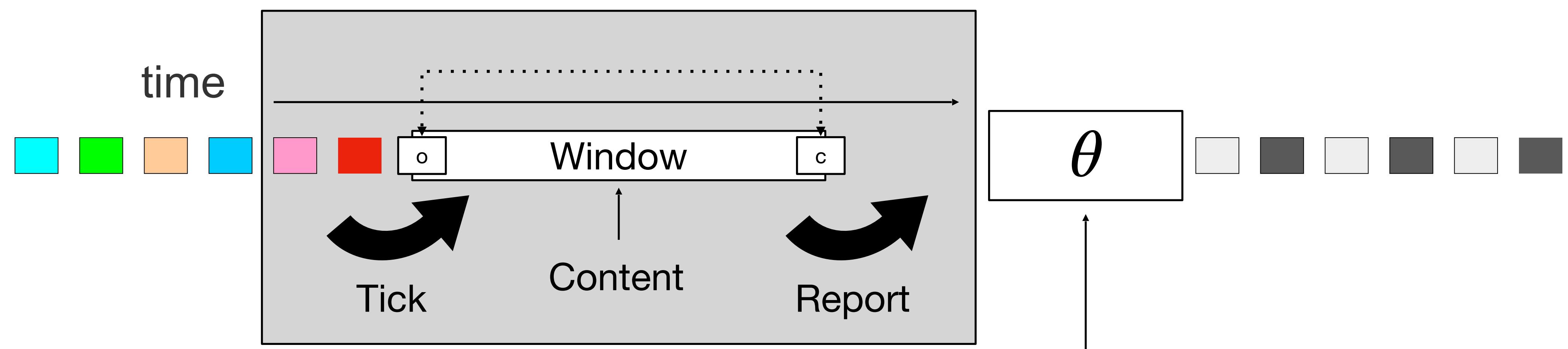
## The Data Flow Model

- Describe how the data system behave over time
  - **What** results are being computed
  - ~~Where in event time are being computed~~
  - **When** in processing time are being materialised
  - ~~How “earlier results” relate to “later refinements”~~



# Execution Semantics

SECRET



Query  
Operator  
(R2R+R2S)

Almost Done!!

Stream  
Processing

Almost!!

Stream  
Processing

Oh Yes!

Stream  
Processing

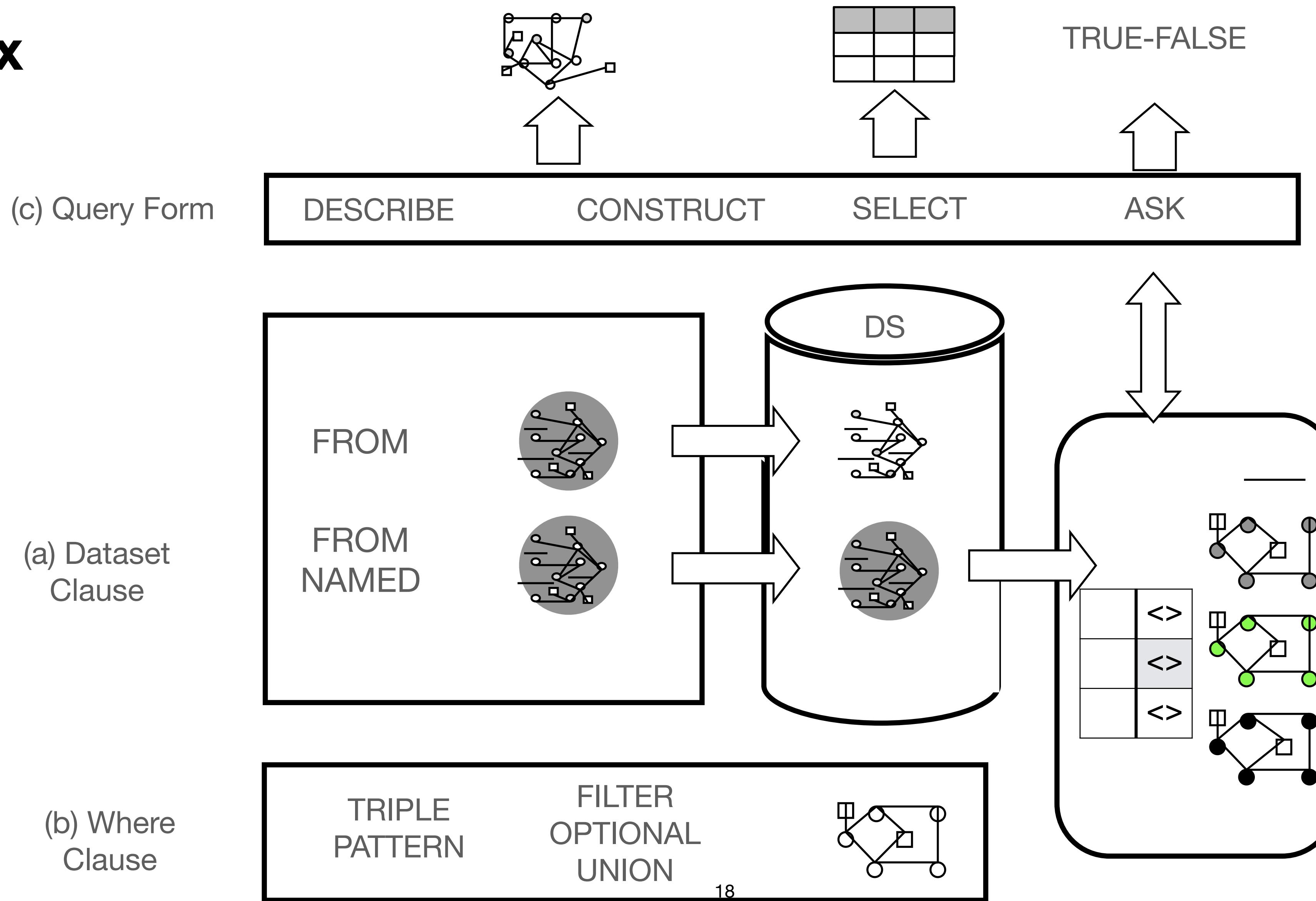
Oh  
Man!

Stream  
Processing

RDF Stream  
Processing

# Do You Recall SPARQL?

## Syntax



# RSP-QL

## Syntax

(c) Query Form

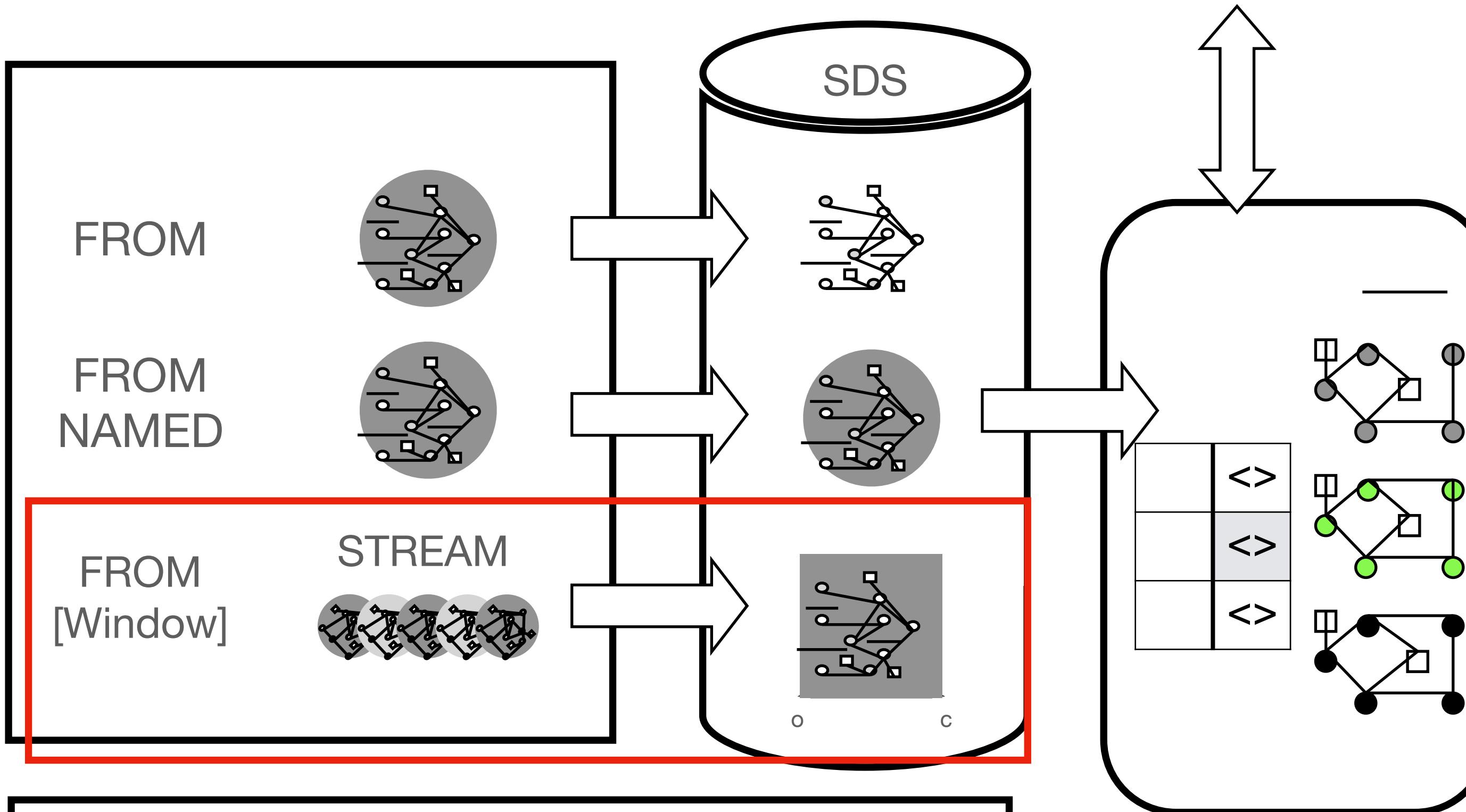
DESCRIBE

CONSTRUCT

SELECT

TRUE-FALSE

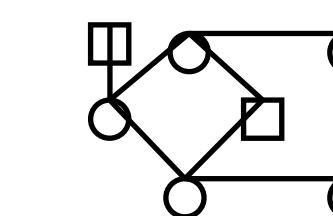
(a) Dataset  
Clause



(b) Where  
Clause

TRIPLE  
PATTERN

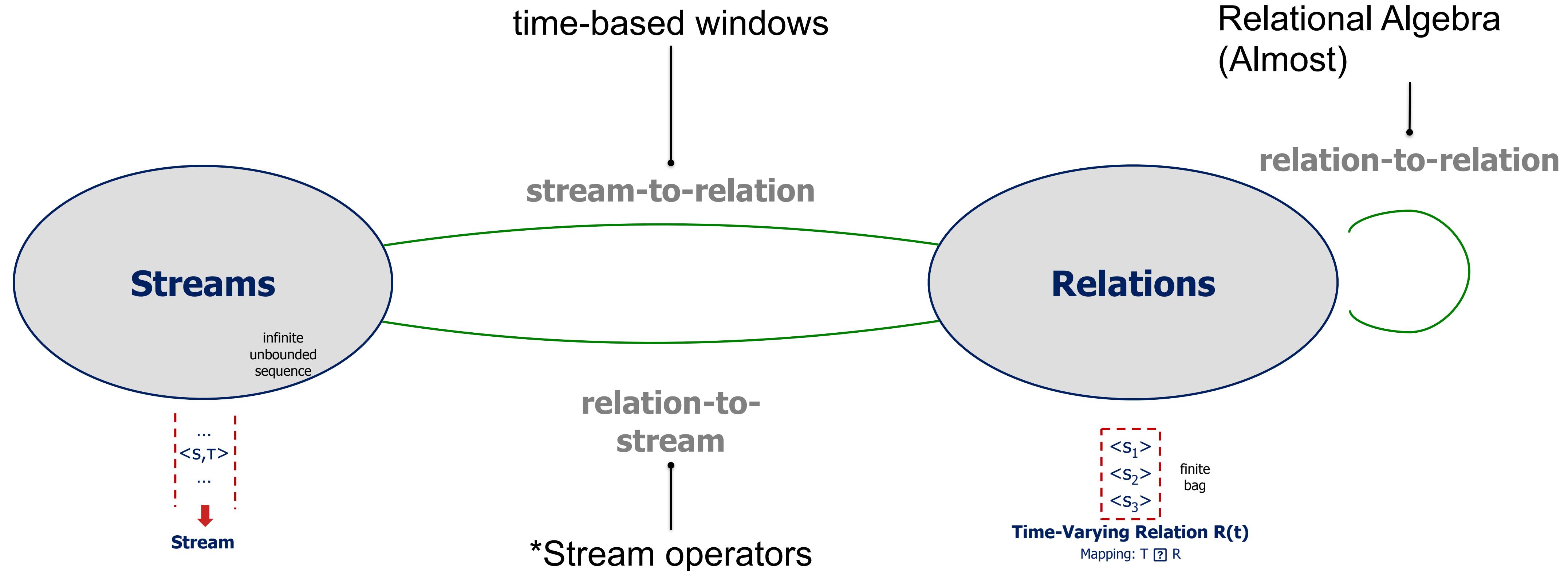
FILTER  
OPTIONAL  
UNION



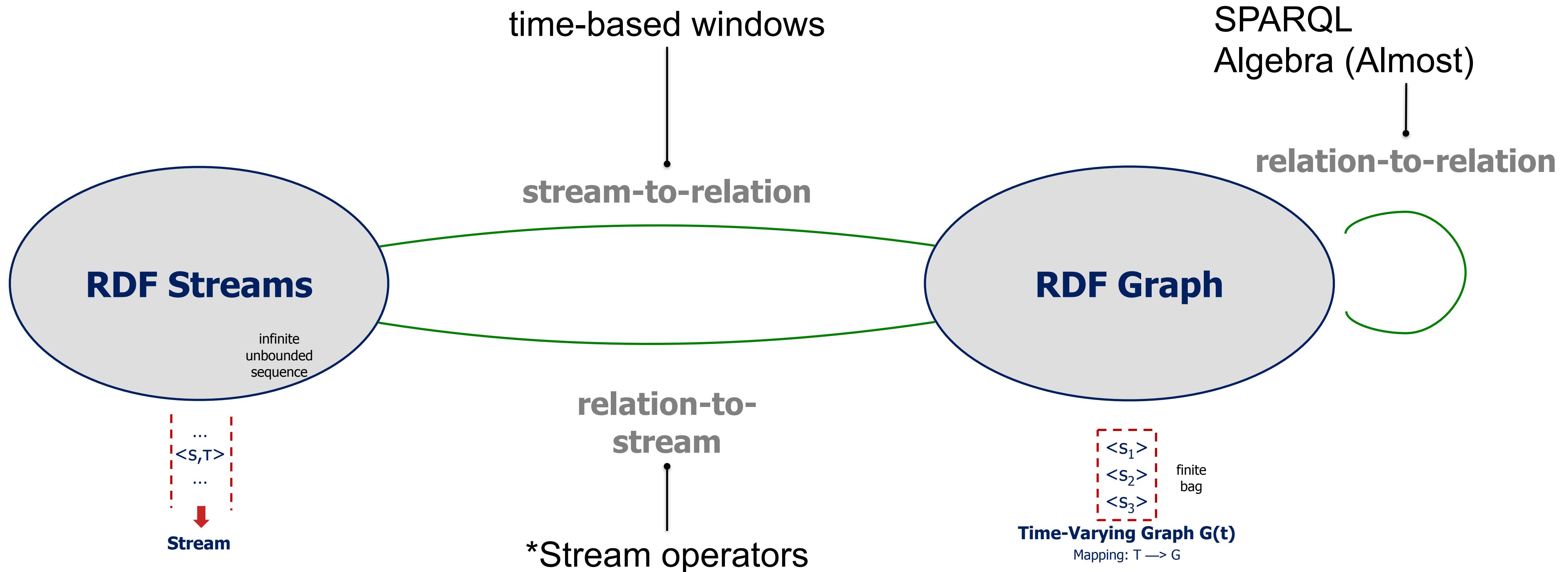
# RSP-QL Query Example

- Using the social stream fb, Who is where?
- **PREFIX :L<http://rwsr/onto#>**  
**REGISTER RSTREAM <WHolsWhereOnFB> AS**  
**SELECT ?room ?person1**  
**FROM NAMED WINDOW <w1> ON <<http://..../fb>> [RANGE PT1S STEP PT1S]**  
**WHERE {**  
    **WINDOW <w1> {**  
        **?person :posts [ :who ?person1 ;**  
            **:where ?room . ]**  
    **}**  
**}**

# The CQL: Families of Operators

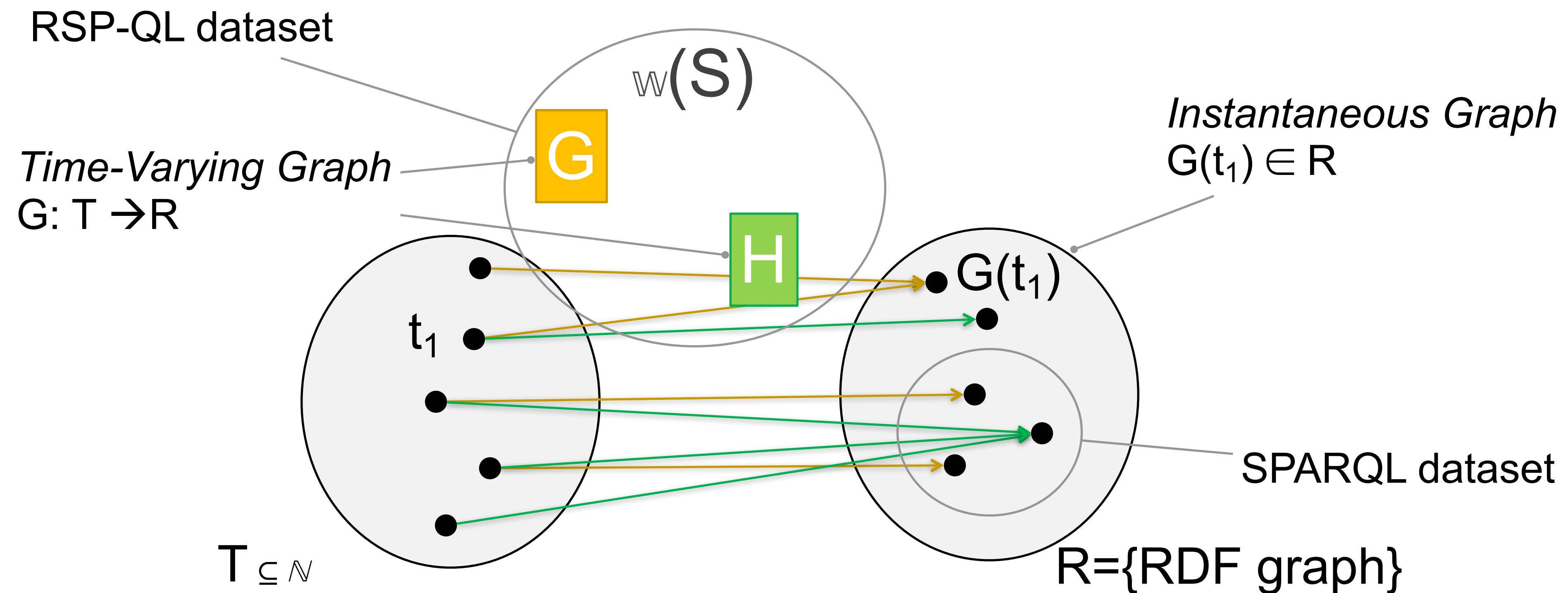


# The RSP-QL: Operators



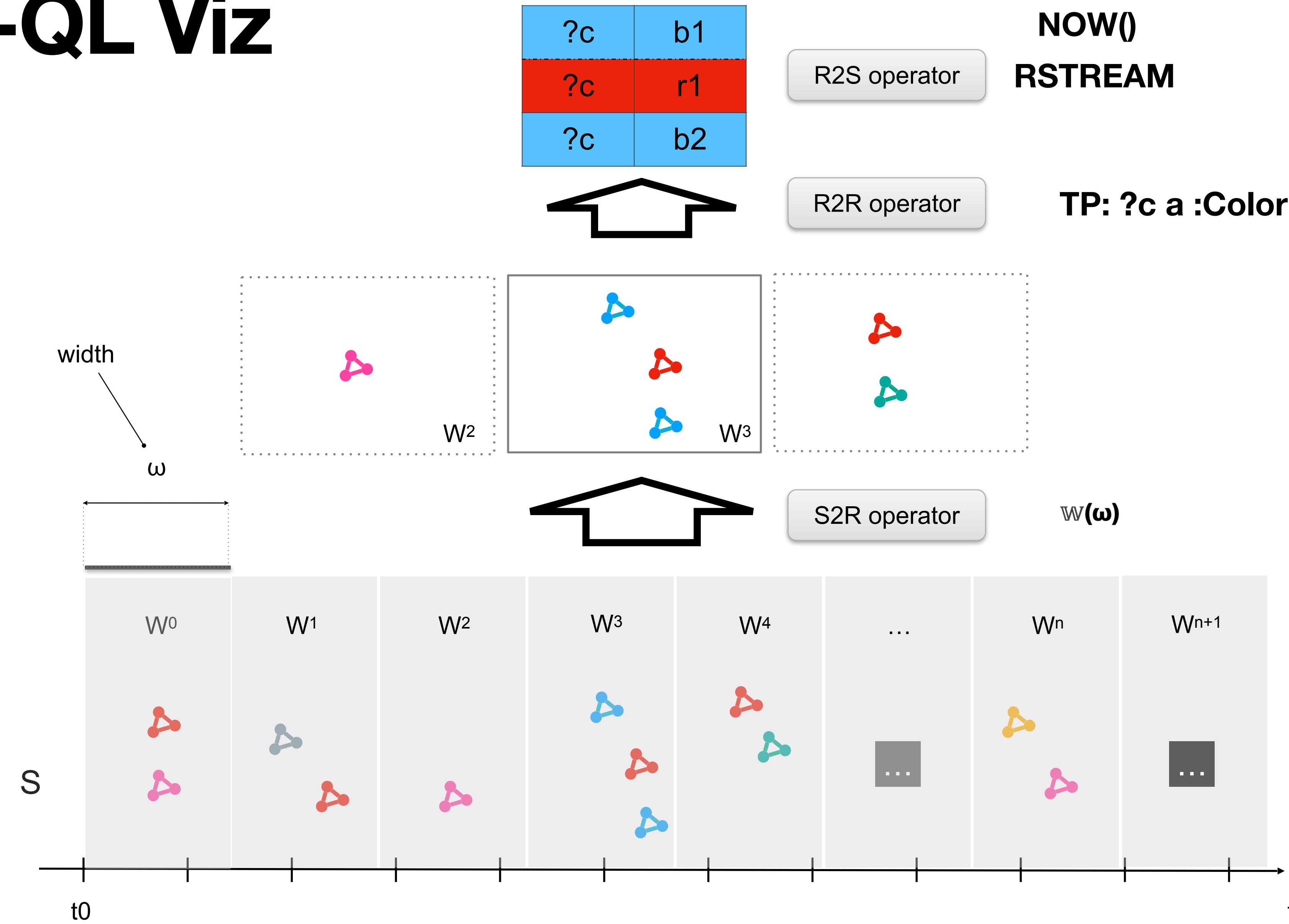
# RSP-QL

## Abstractions



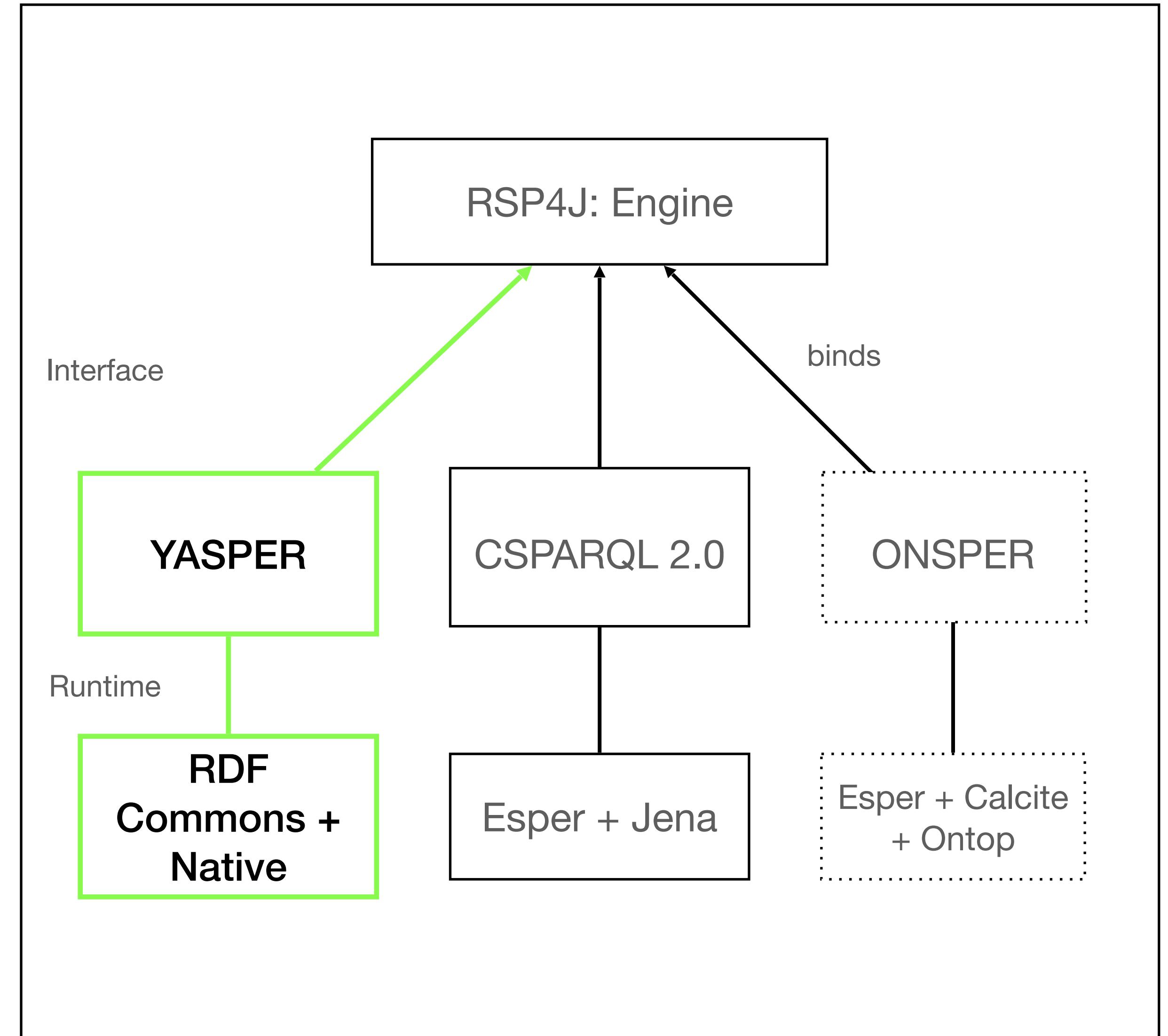
# RSP-QL Viz

## S2R



# RSP4J

- Allows controlling the RSP Engine, e.g., stream ingestion, query registration, result stream.
- Can reproduce the execution semantics of common RSP engines
  - Includes some bindings, YASPER, CSPARQL 2.0, ONSPER\*



# Continuous Program

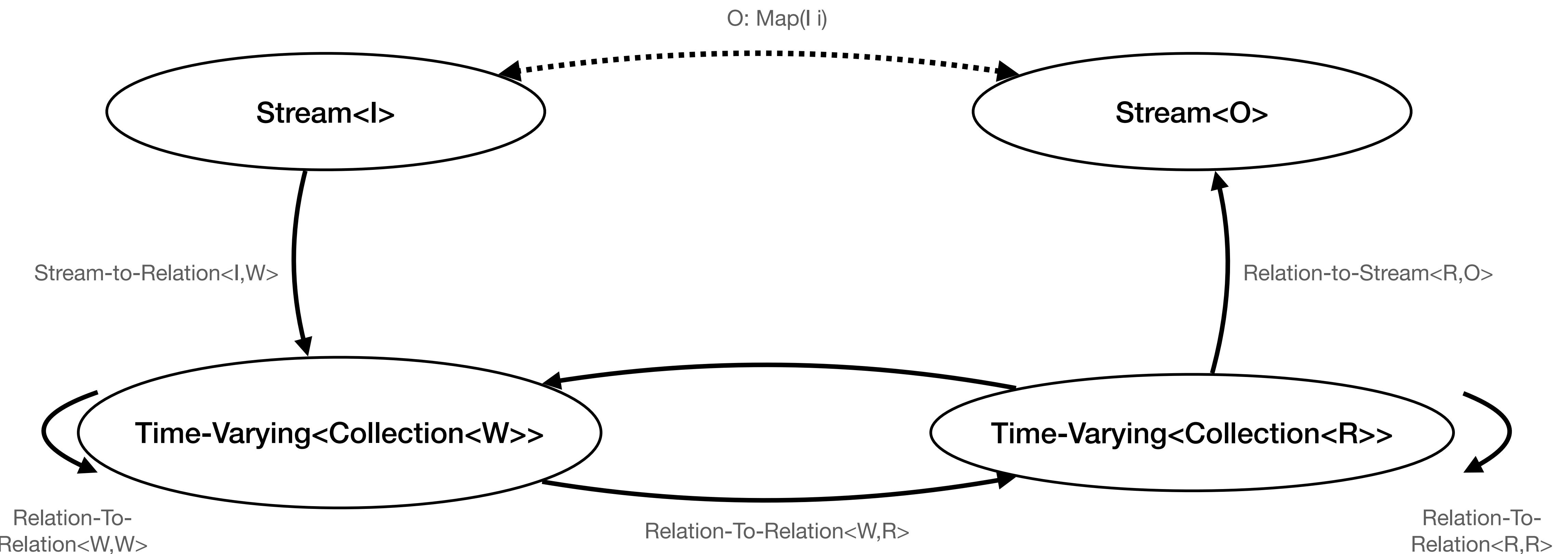
- A continuous program evaluates a set of (stateful) functions (aka Tasks) on an input data stream, then it populates an output stream with the computation.
- In RSP4J, a continuous program is **described by four data types**, i.e.
  - **I**, i.e., the type of the data items in the **input data stream**
  - **O**, i.e., the Type of the data items in the **output data stream**
  - **W**, and **R**, which respectively describe the **intermediate data types**, right **after windowing** and after **the application of an R2R Operator**.



Do you think those are tables you're querying?

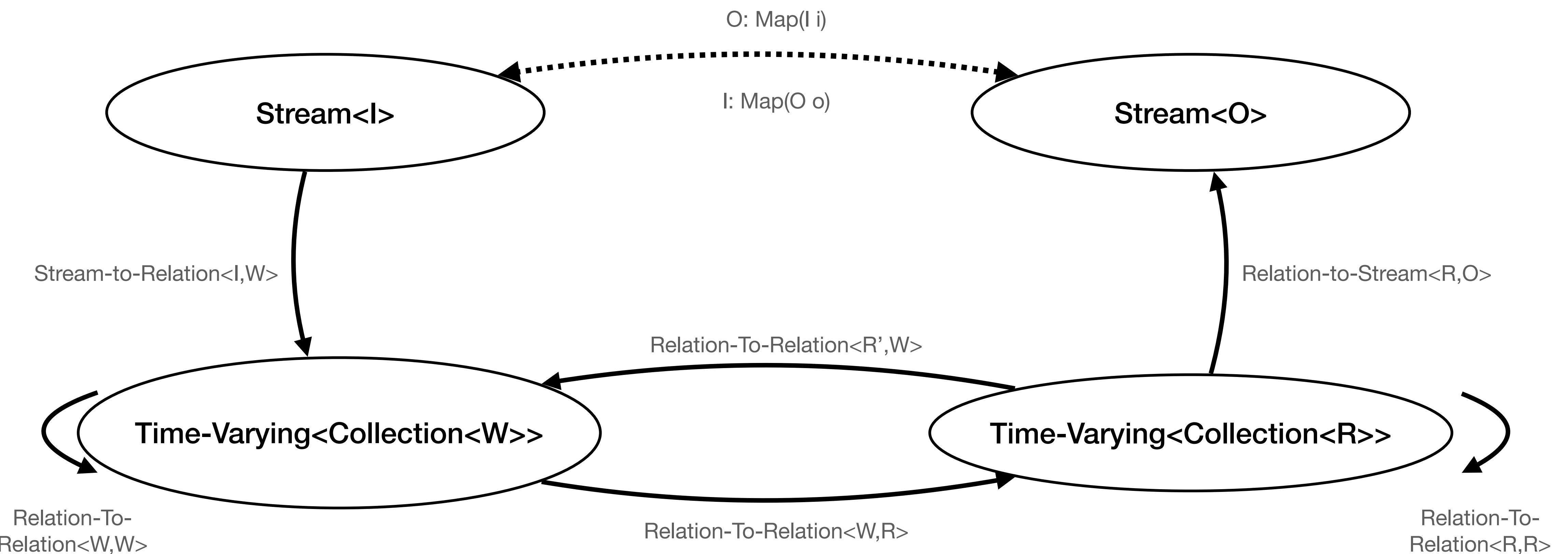
# RSP4J

## Generics



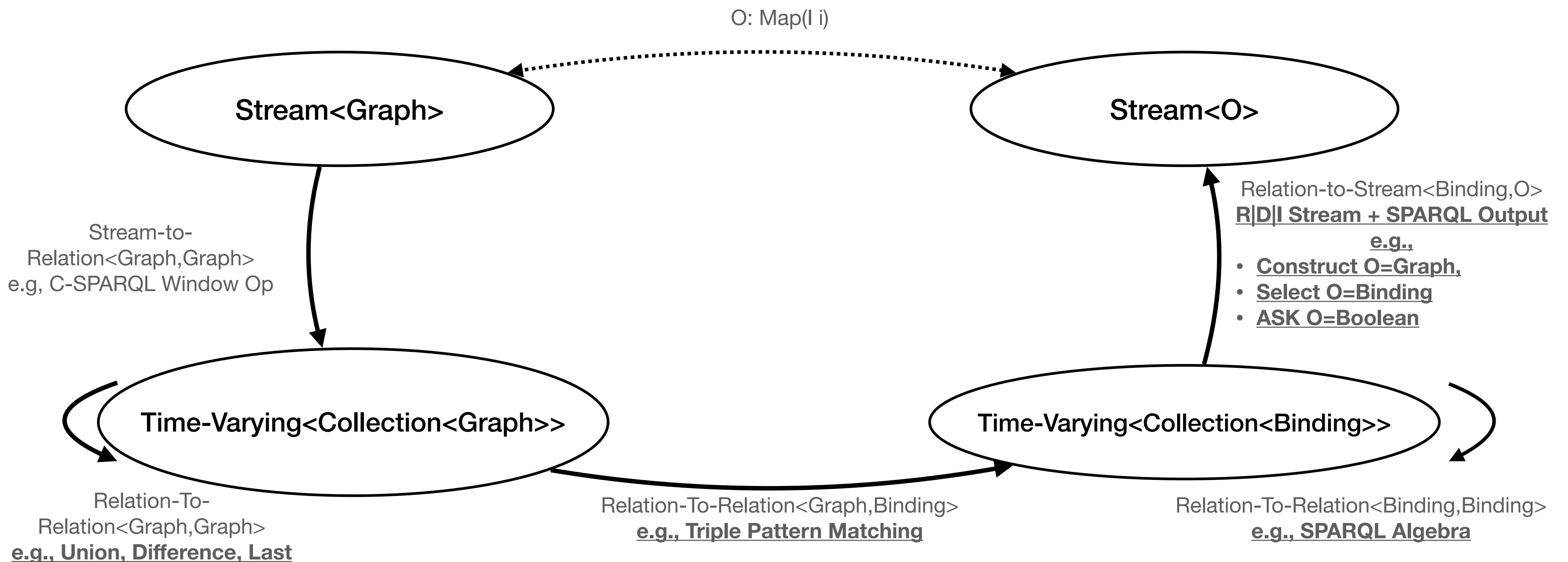
# RSP4J

## Generics



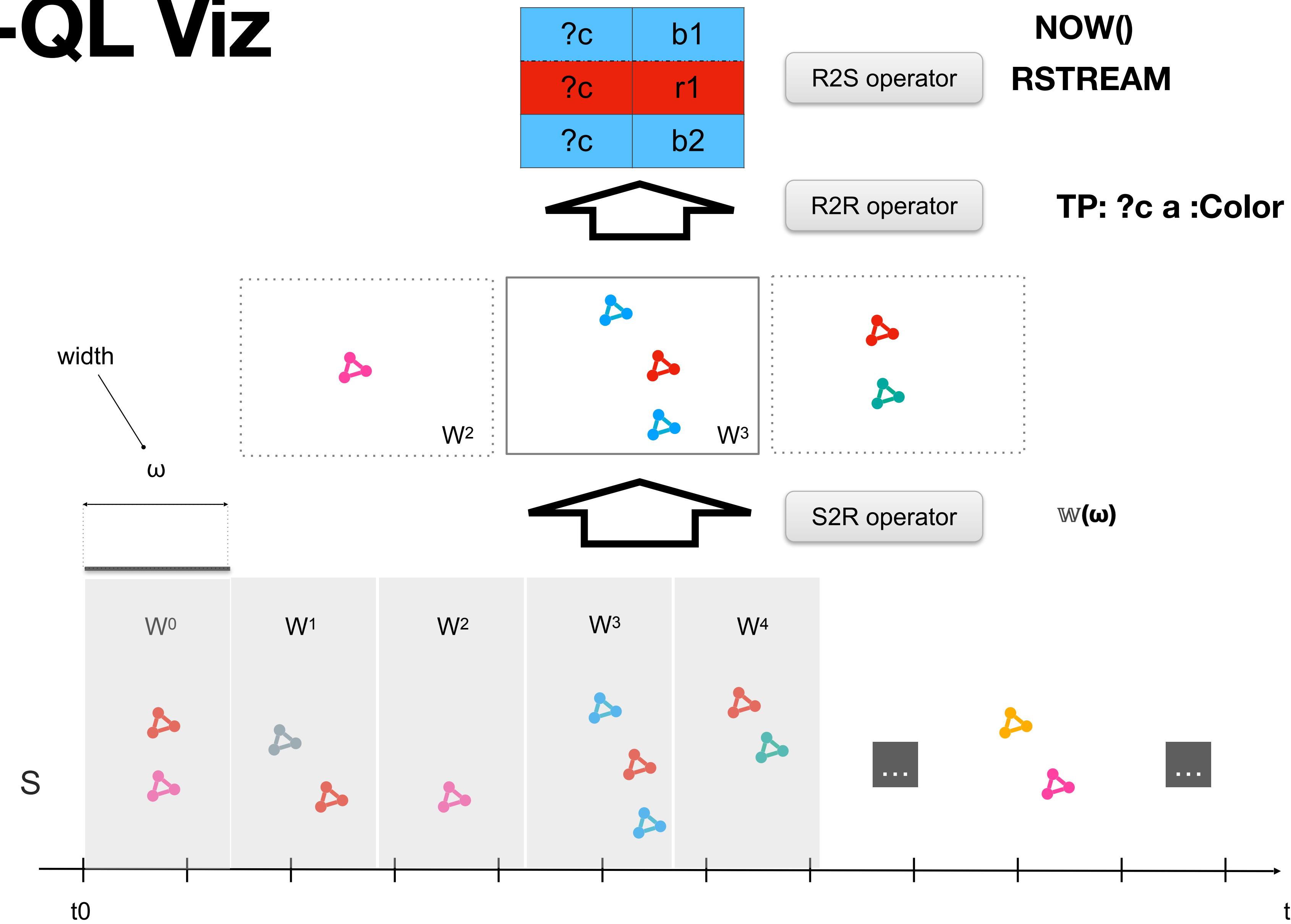
# RSP4J

## RSP-QL Configuration



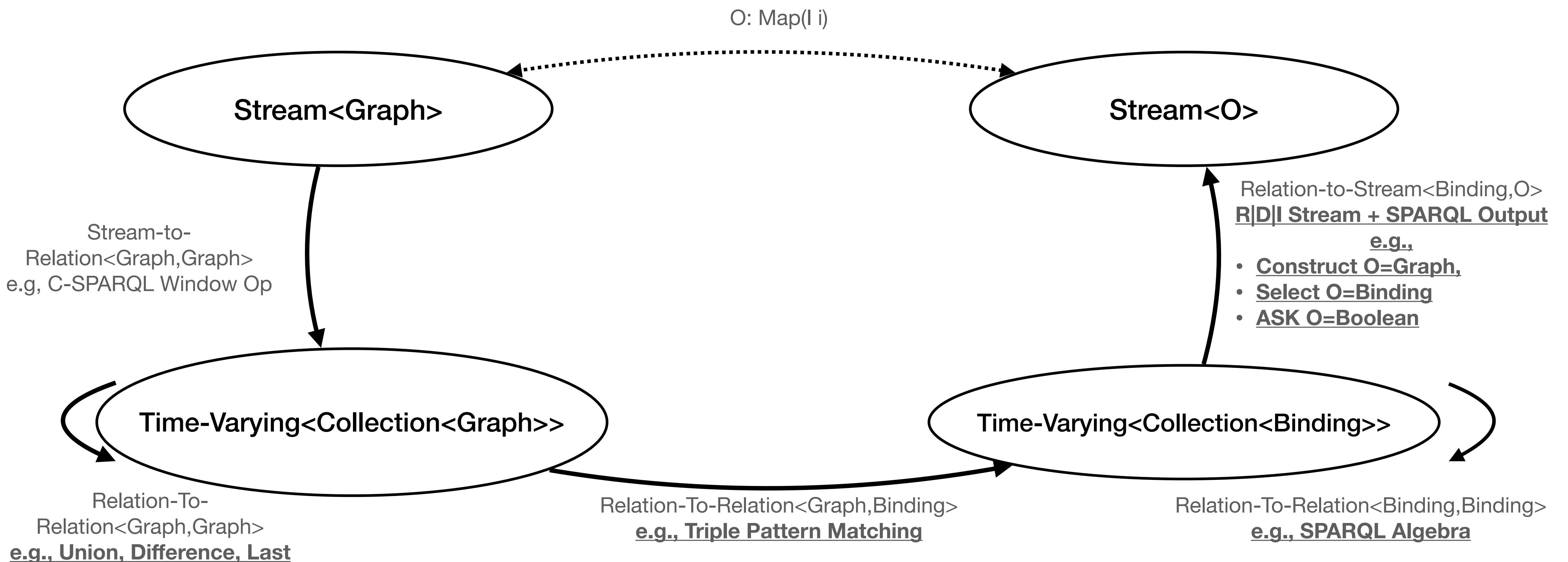
# RSP-QL Viz

## S2R



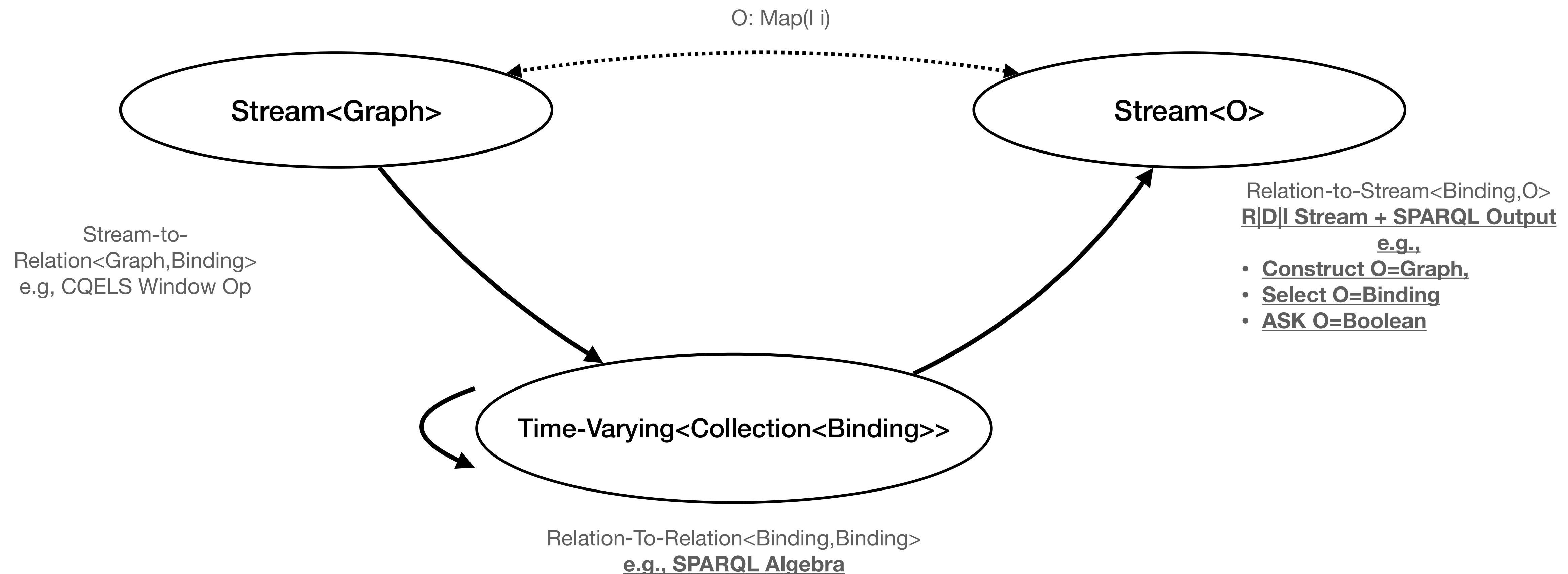
# RSP4J

## RSP-QL Configuration



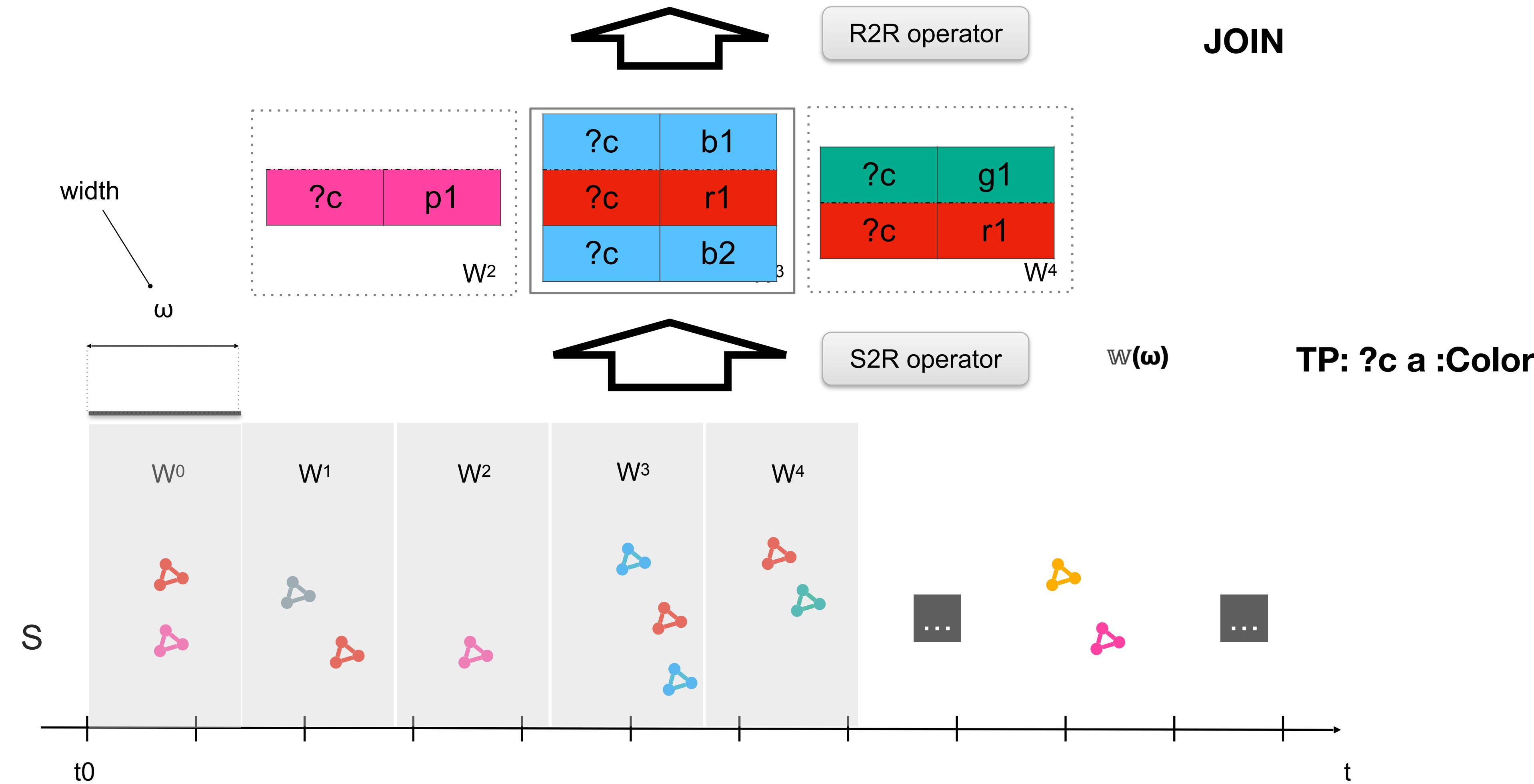
# RSP4J

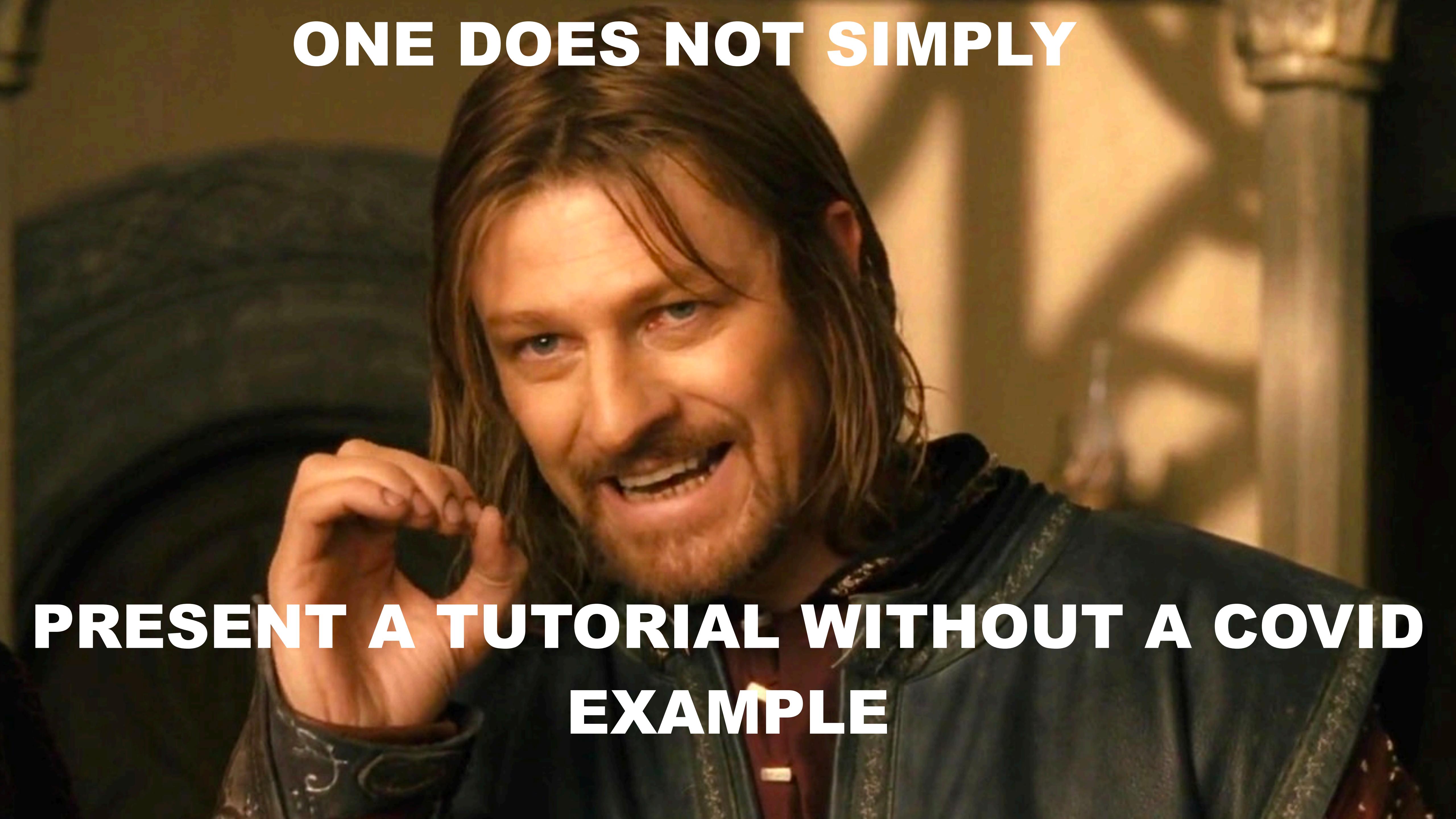
## RSP-QL Configuration



# RSP-QL Viz

## S2R

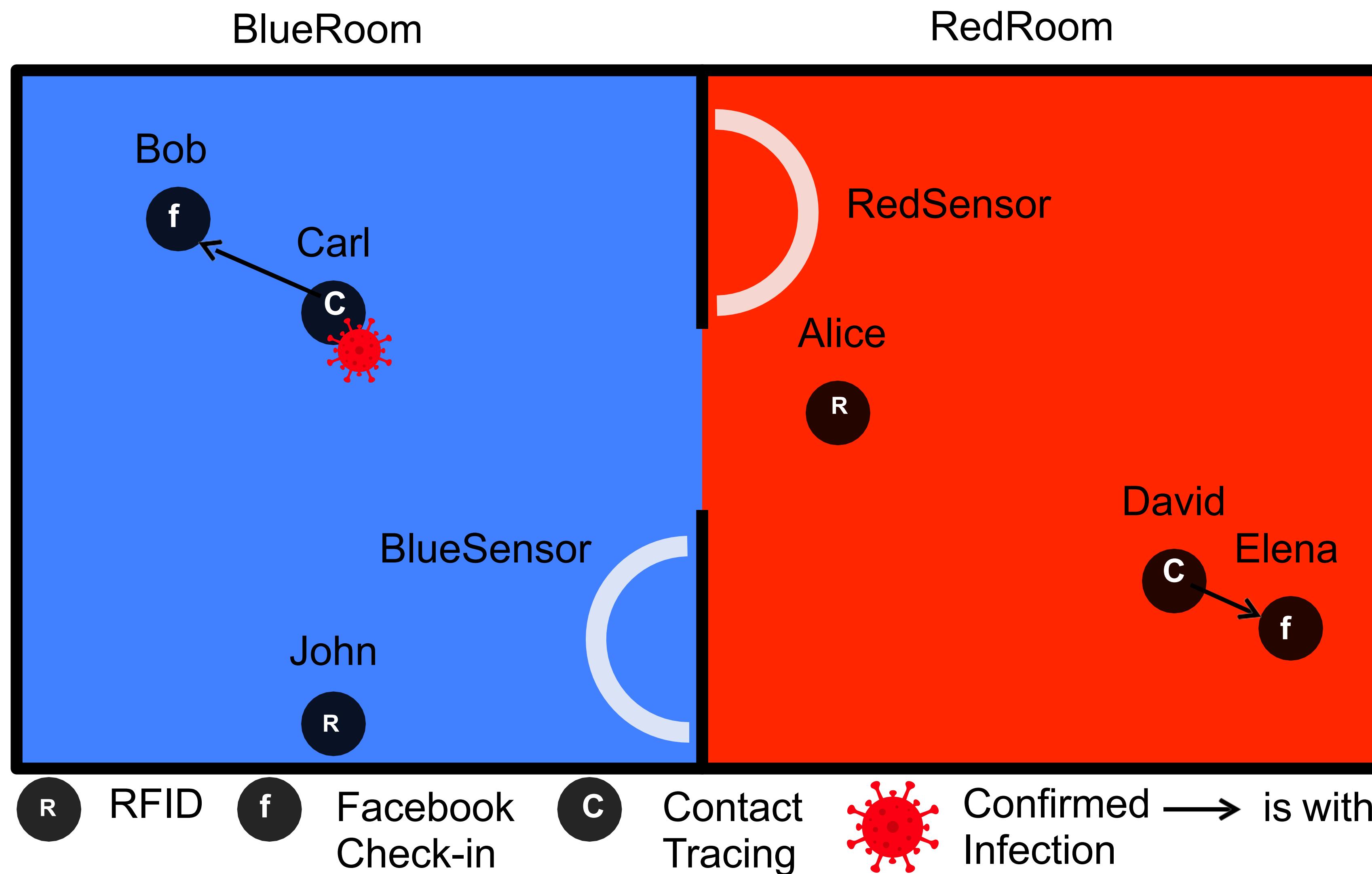


A close-up of Aragorn's face from The Lord of the Rings. He has long brown hair and a beard, looking slightly to the side with a serious expression. He is wearing a dark leather vest over a tunic.

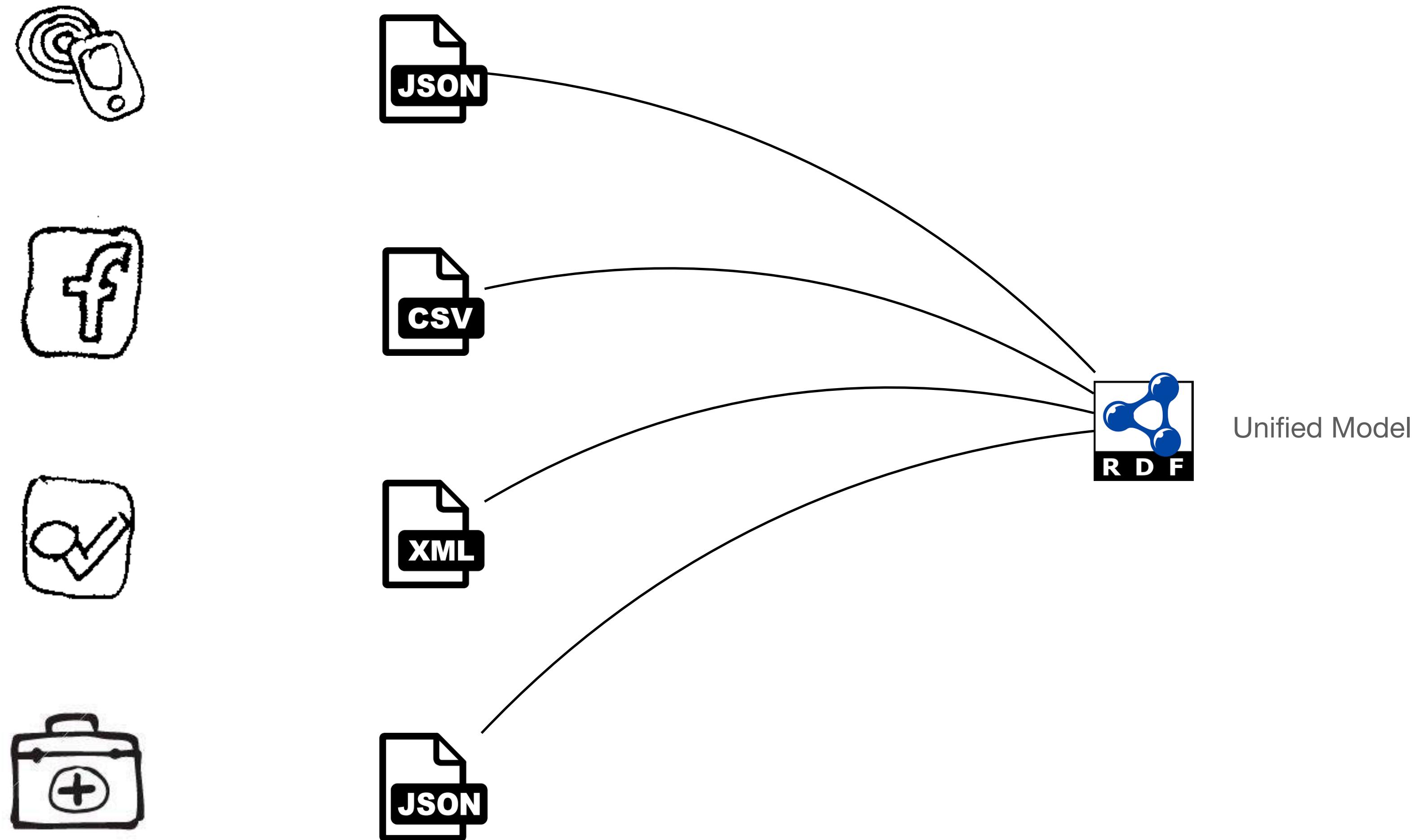
ONE DOES NOT SIMPLY

PRESENT A TUTORIAL WITHOUT A COVID  
EXAMPLE

# Covid Scenario



# Realistic example data in the streams



# Simplified example data in the streams



Sensor	Room	Person	Time-stamp
RedSensor	RedRoom	Alice	T <sub>1</sub>
...	...	...	...



Person	ChecksIn	Time-stamp
Bob	BlueRoom	T <sub>2</sub>
...	...	...



Person	With	Time-stamp
Carl	Bob	T <sub>2</sub>
...	...	...



Person	Result	Time-stamp
Carl	Positive	T <sub>3</sub>
...	...	...

# Simplified example data in the streams



<b>Sensor</b>	<b>Room</b>	<b>Person</b>	<b>Time-stamp</b>
RedSensor	RedRoom	Alice	T <sub>1</sub>
...	...	...	...

:observationX a :RFIDObservation .  
 :observationX :who :Alice .  
 :observationX :where :RedRoom .  
 :Alice :isIn :RedRoom .



<b>Person</b>	<b>ChecksIn</b>	<b>Time-stamp</b>
Bob	BlueRoom	T <sub>2</sub>
...	...	...

:postY a :FacebookPost .  
 :postY :who :Bob .  
 :postY :where :BlueRoom .  
 :Bob :isIn :BlueRoom .



<b>Person</b>	<b>With</b>	<b>Time-stamp</b>
Carl	Bob	T <sub>2</sub>
...	...	...

:postZ a :ContactTracingPost .  
 :postZ :who :Carl .  
 :Carl :isWith :Bob .



<b>Person</b>	<b>Result</b>	<b>Time-stamp</b>
Carl	Positive	T <sub>3</sub>
...	...	...

:postQ a :TestResultPost .  
 :postQ :who :Carl .  
 :postQ :hasResult :positive

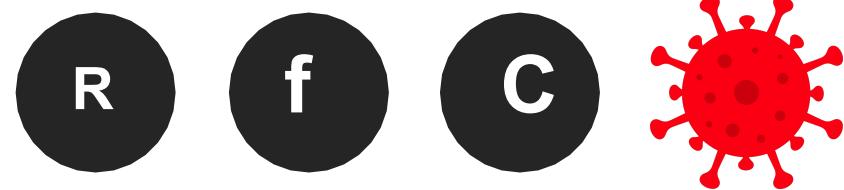
# Covid Scenario Insights

- Who is with who in which room?



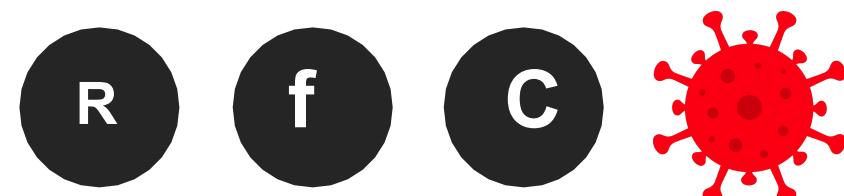
e.g. *Dave, Elena, RedRoom*

- Who is infected through close contact?

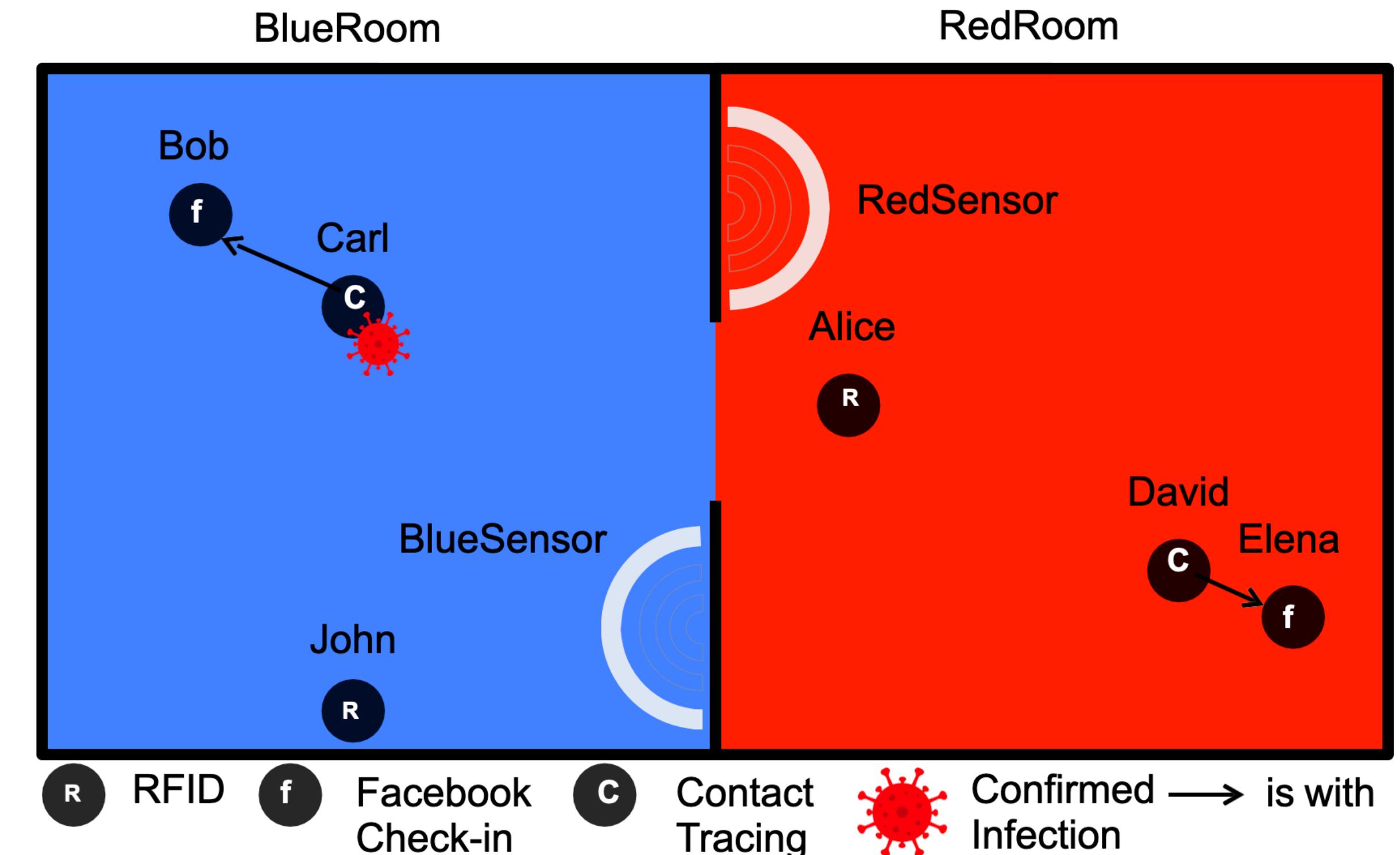


e.g. *Bob, Carl*

- Who is in a room where there is an infected person?



e.g. *BlueRoom, John, Bob, Carl*





Demo Time!

# Are you following? Let's check!

[source [http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/\\_jcr\\_content/flash/image.img.png](http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/_jcr_content/flash/image.img.png) ]

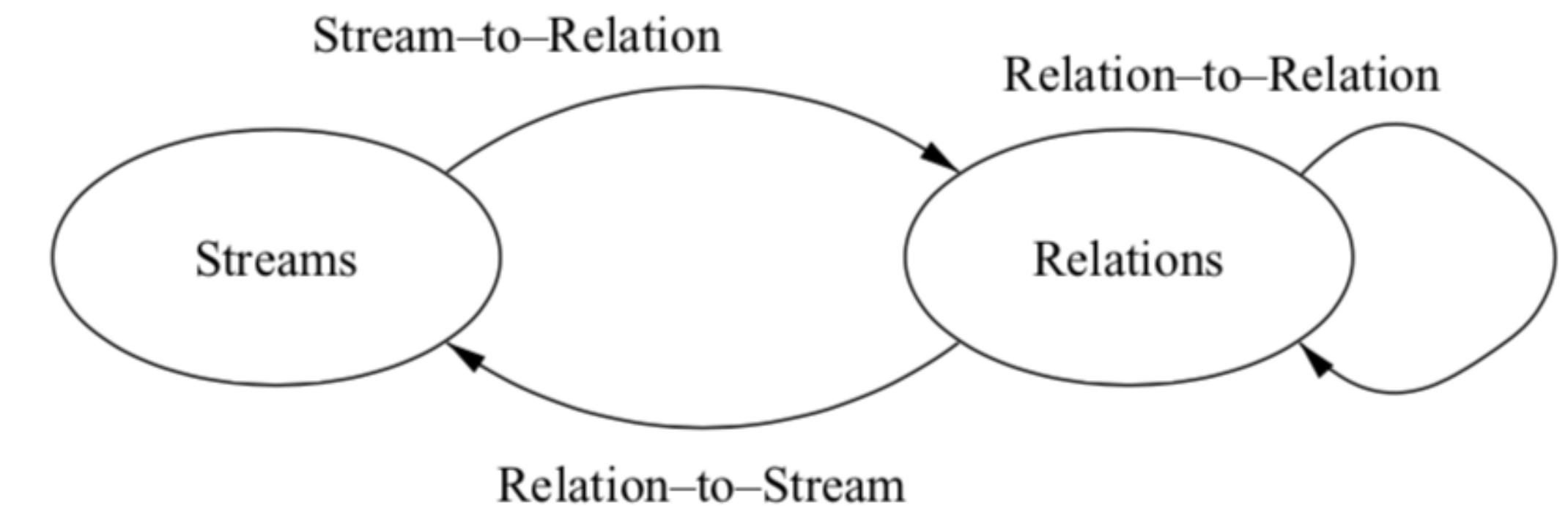
- Open QueryProcessingAssignment  
(in org.streamreasoning.rsp4j.thewebofconf2022.processing.assignment)
- Define a query that extracts who might be infected through close contact in the last 10 minutes, with test results valid for 24 hours.
- Check the example (QueryProcessingExample) to get you started
- Git repo: <https://github.com/pbonete/WSP-TheWebConf2022Tutorial>



# RSP4J

## 10.000ft View: Operators

- Generalize CQL Style of operators families
- Includes already canonical examples of S2R
  - CQELS's Window Operator
  - CSPARQL's Window Operator
- Includes custom implementation of R2R
  - BGP
  - TP
- Includes generic implementation of R2S
  - RStream, IStream, DStream



# RSP4J

## Operators

- Customize Tasks using TaskBuilder
  - Specify operators (S2R, R2R, R2S)
  - Optional aggregation
  - Creation of ContinuousProgram from Task



Demo Time!

# Are you following? Let's check!

- Open AbstractionAssignment  
(in org.streamreasoning.rsp4j.thewebsite2022.processing.assignment)
- Follow the instructions and define the different operators in the Task
- Check the example (AbstractionExample) to get you started



[source [http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/\\_jcr\\_content/flash/image.img.png](http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/_jcr_content/flash/image.img.png) ]



Demo Time!

# Are you following? Let's check!

- Open CustomR2RAssignment  
(in org.streamreasoning.rsp4j.thewebsite2022.processing.assignment)
- Follow the instructions and create your own R2R operator
- Check the example (CustomR2RExample) to get you started



[source [http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/\\_jcr\\_content/flash/image.img.png](http://origin-www.yoursingapore.com/content/traveller/zh/browse/see-and-do/hands-on/_jcr_content/flash/image.img.png) ]



# Web Stream Processing with RSP4J

The Web Conf 2022

Pieter Bonte (BE) & Matteo Belcao (IT) & Marco Balduini (IT) & Emanuele Della Valle (IT)

26/04/2022