



GHENT
UNIVERSITY

imec



POLITECNICO
MILANO 1863



QUANTIA
consulting

WEB STREAM PROCESSING

WITH RSP4J AND ONTOPSTREAM

TheWebConf 2022, Online, hosted by Lyon, France - 26-4-2022

PIETER BONTE

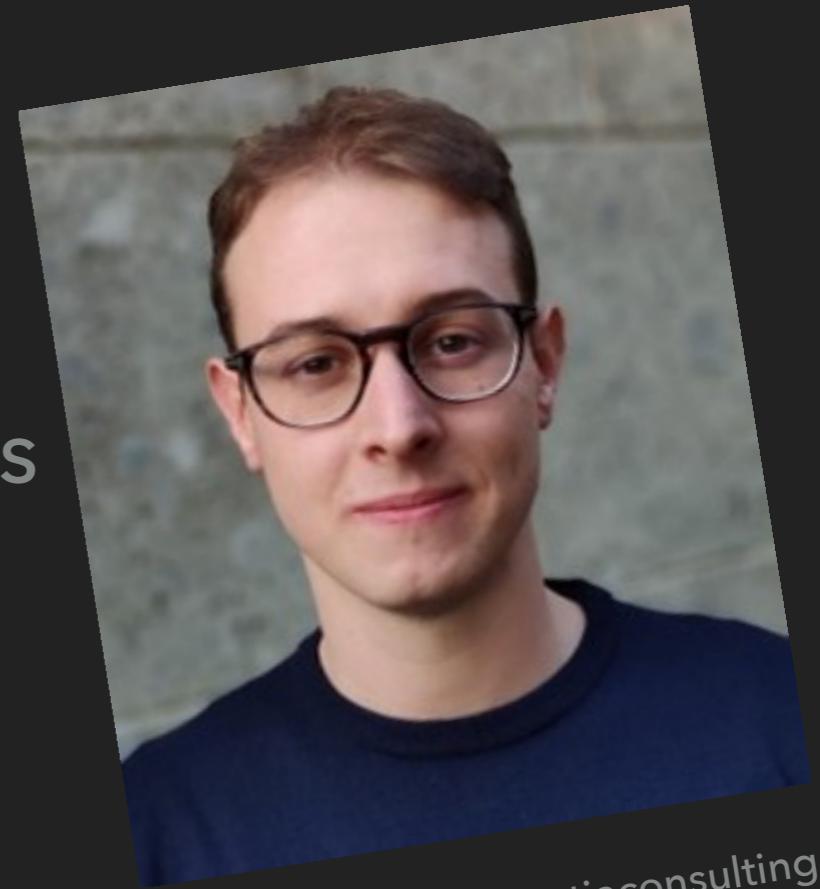
- ▶ Postdoctoral researcher at Ghent University - imec
- ▶ Expert in graph stream processing, data inference and semantic technologies
- ▶ Maintainer of RSP4J, author of C-SPRITE & Streaming MASSIF
- ▶ ~8 years experience in innovation and research projects



pieter.bonte@ugent.be
<http://pieterbonte.be/>

MATTEO BELCAO

- ▶ Data Engineer at Quantia Consulting
- ▶ Expert in Ontology Based Data Access and Stream Processing technologies
- ▶ author of Chimera Suite
- ▶ 2+ years of experience in research and innovation projects



matteo.belcao@quantiaconsulting.com
<http://www.quantiaconsulting.com>

EMANUELE DELLA VALLE

- ▶ Associate Professor at DEIB
Politecnico di Milano
- ▶ Expert in semantic technologies
and stream computing
- ▶ Brander of **stream reasoning**
- ▶ 20+ years of experience in research
and innovation projects
- ▶ Startupper



emanuele.dellavalle@polimi.it
<http://emanueledellavalle.org>

PRELIMINARY SETUP

- ▶ During this introduction, you can get ready for the hands-on sessions:
- ▶ Install docker and docker-compose:
 - ▶ docker: <https://docs.docker.com/engine/install/>
 - ▶ docker-compose: <https://docs.docker.com/compose/install/>
- ▶ Clone our repository:
<https://github.com/pbonte/WSP-TheWebConf2022Tutorial>
 - ▶ e.g., `git clone https://github.com/pbonte/WSP-TheWebConf2022Tutorial.git`
 - ▶ Also downloading the zip is ok

BIG DATA TECHS CAN TAME VOLUME

- ▶ Hadoop, MapReduce, HIVE
- ▶ “schema on read” methodology
- ▶ spark (x100 faster)
- ▶ “data lake” concept



BIG DATA TECHS CAN TAME VELOCITY

- ▶ Storm
- ▶ Kafka
- ▶ Spark Streaming
- ▶ Flink
- ▶ paradigmatic change
 - ▶ from persistent data and transient queries
 - ▶ **to persistent queries and transient data**

BIG DATA TECHS CANNOT TAME VOLUME AND VELOCITY SIMULTANEOUSLY



BIG DATA TECHS CAN TAME VARIETY USING SEMANTIC WEB TECHNOLOGIES

- ▶ RDF data model
- ▶ SPARQL query language
- ▶ OWL ontological language
- ▶ R2RML mapping language
- ▶ Ontology Based Data Access methodology

VARIETY & VERACITY MAKES PROBLEMS HARDER



STILL THERE ARE USERS WHOSE DECISIONS NEED TO TAME ALL Vs

WEB STREAM PROCESSING FOR SMART CITIES



- ▶ Can you **suggest where to spend my next hours** given my interests, the presence of people and what they're doing?
-
- ▶ **100,000s people** leaving **10,000s digital footprint per second** via Call Data Records, Bluetooth, WiFi, Social Media, ...

REQUIREMENT ANALYSIS

A system able to answer those queries must be able to

- ▶ handle **massive** datasets
- ▶ process **data streams** on the fly
- ▶ cope with **heterogeneous** datasets
- ▶ cope with **incomplete** data
- ▶ cope with **noisy** data
- ▶ provide **reactive answers**
- ▶ support **fine-grained information access**
- ▶ integrate **complex domain models**

	Volume	Velocity	Variety	Veracity
x				
	x			
		x		
		x	x	
			x	
		x		
		x	x	
			x	

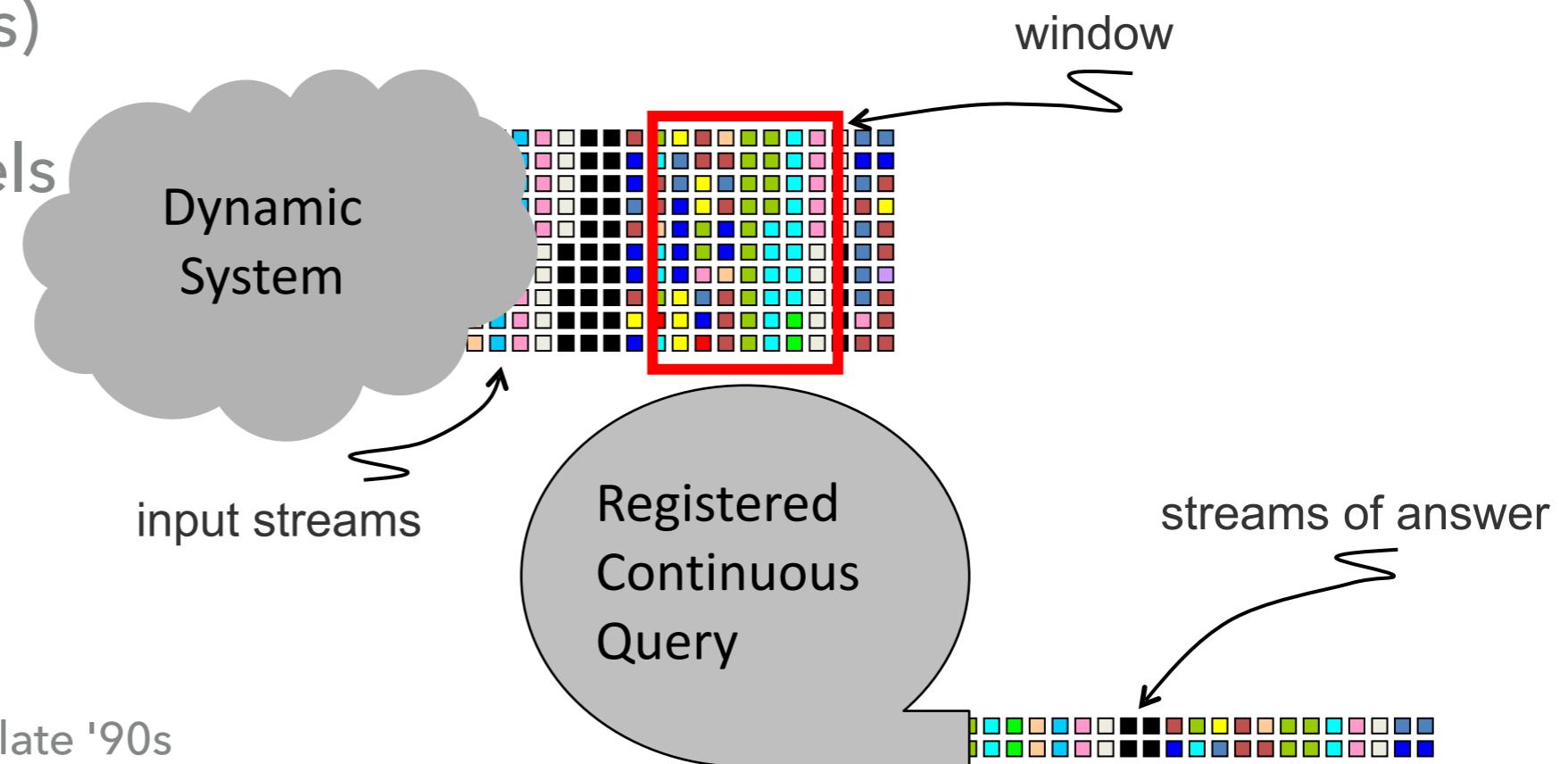
DATA STREAMS

- ▶ Data Streams are usually **unbounded**
- ▶ **No assumption** can be made **on** data arrival **order**
- ▶ Data items in streams often represent **observations not facts**
- ▶ Size and time constraints make it **difficult to store** and process data stream elements **after their arrival**
- ▶ **One-time processing** is the typical mechanism used to deal with streams



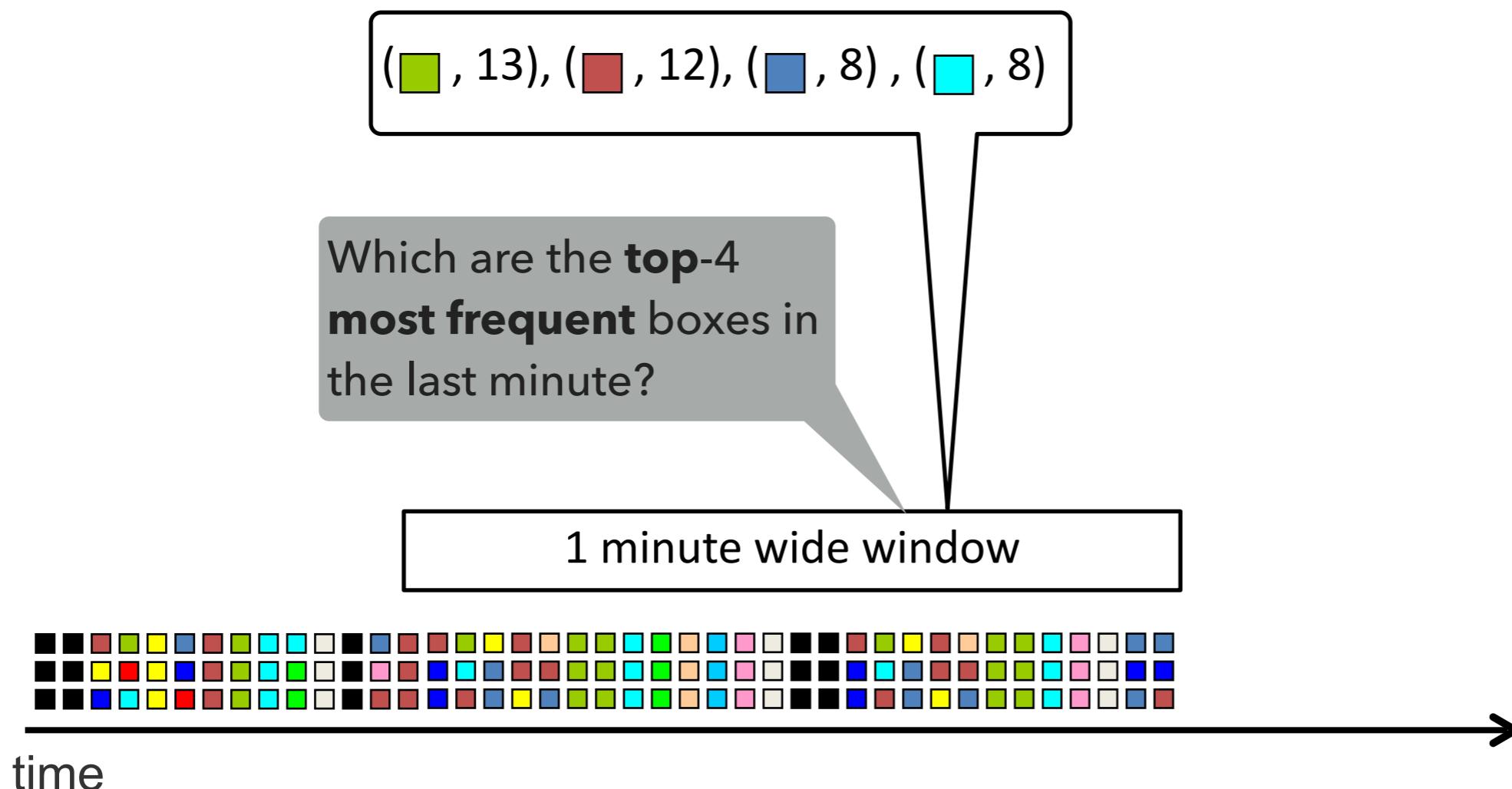
THE PARADIGMATIC CHANGE* OF STREAM PROCESSING

- ▶ From **persistent data** and **transient queries**
(one time semantics)
- ▶ To **transient data** and **persistent queries**
(continuous semantics)
- ▶ Two competing models
 - ▶ DSMS
 - ▶ CEP



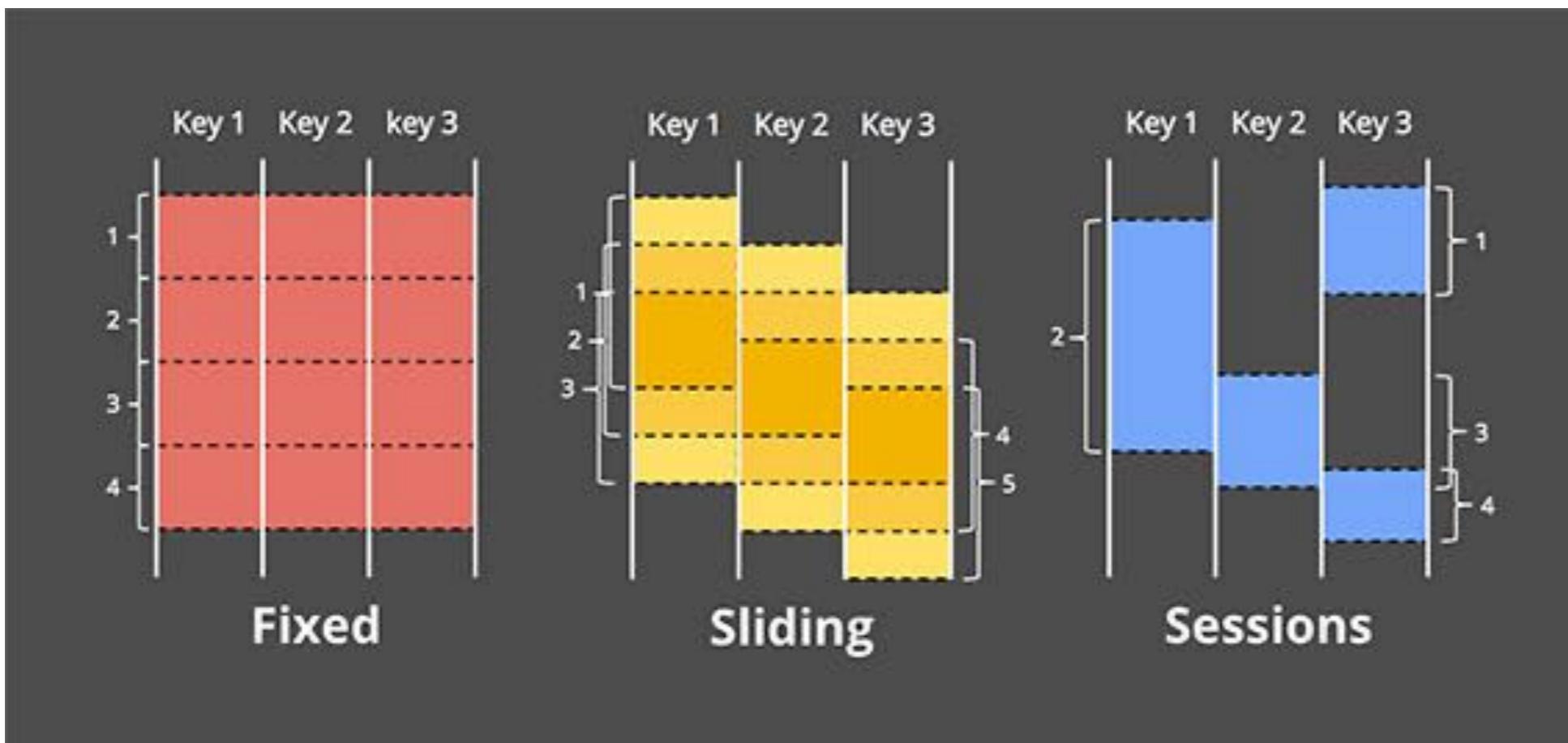
*first arose in DB community in the late '90s

STREAM PROCESSING A USER PERSPECTIVE



WINDOWS

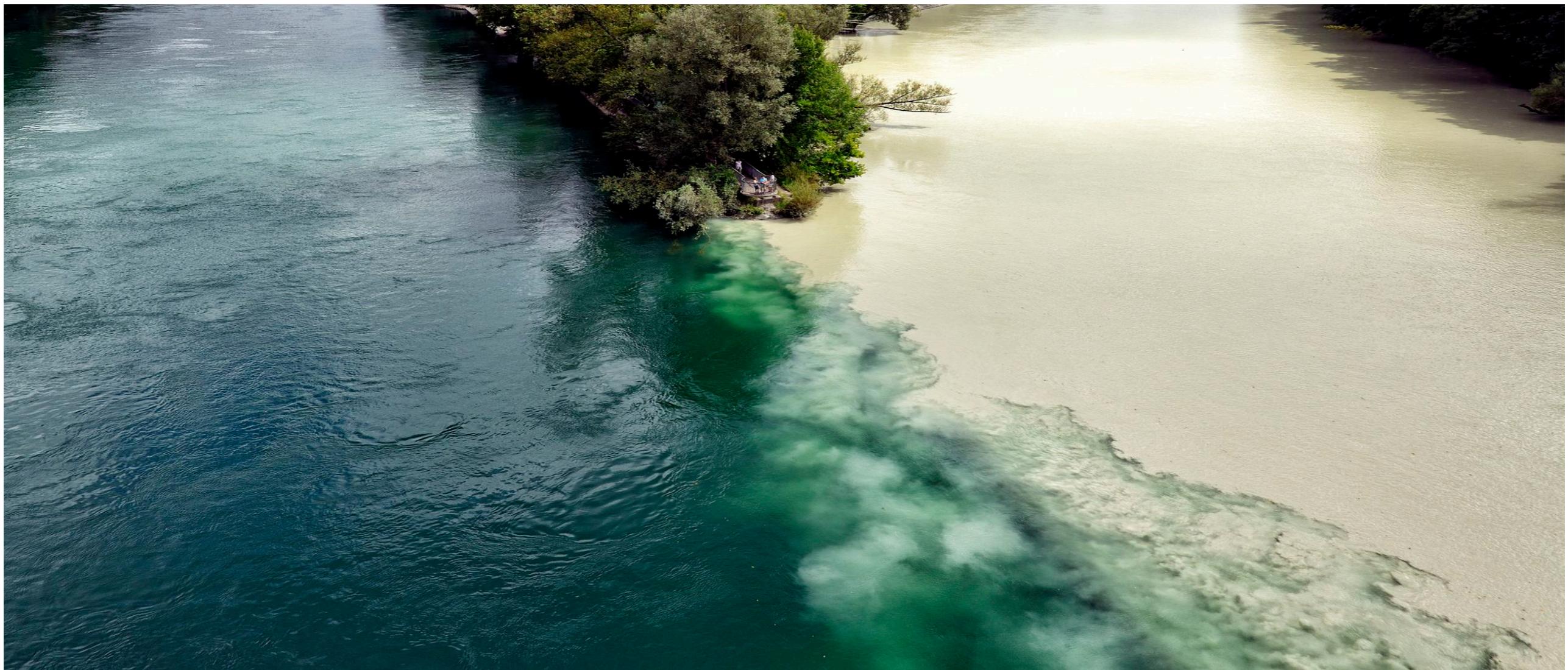
- ▶ Windows define a finite sub-streams of an unbounded stream



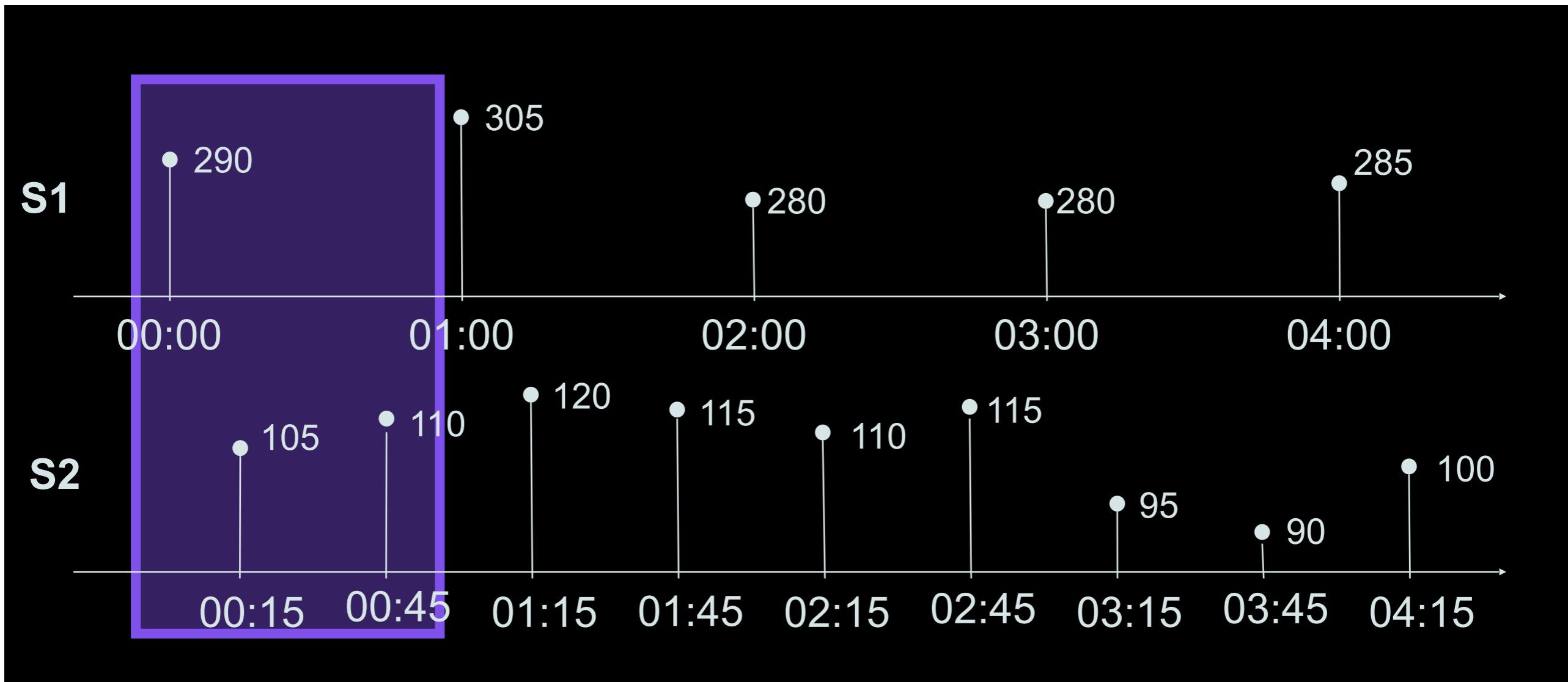
- ▶ They can be interpreted as locally closed worlds

STATE-OF-THE-ART

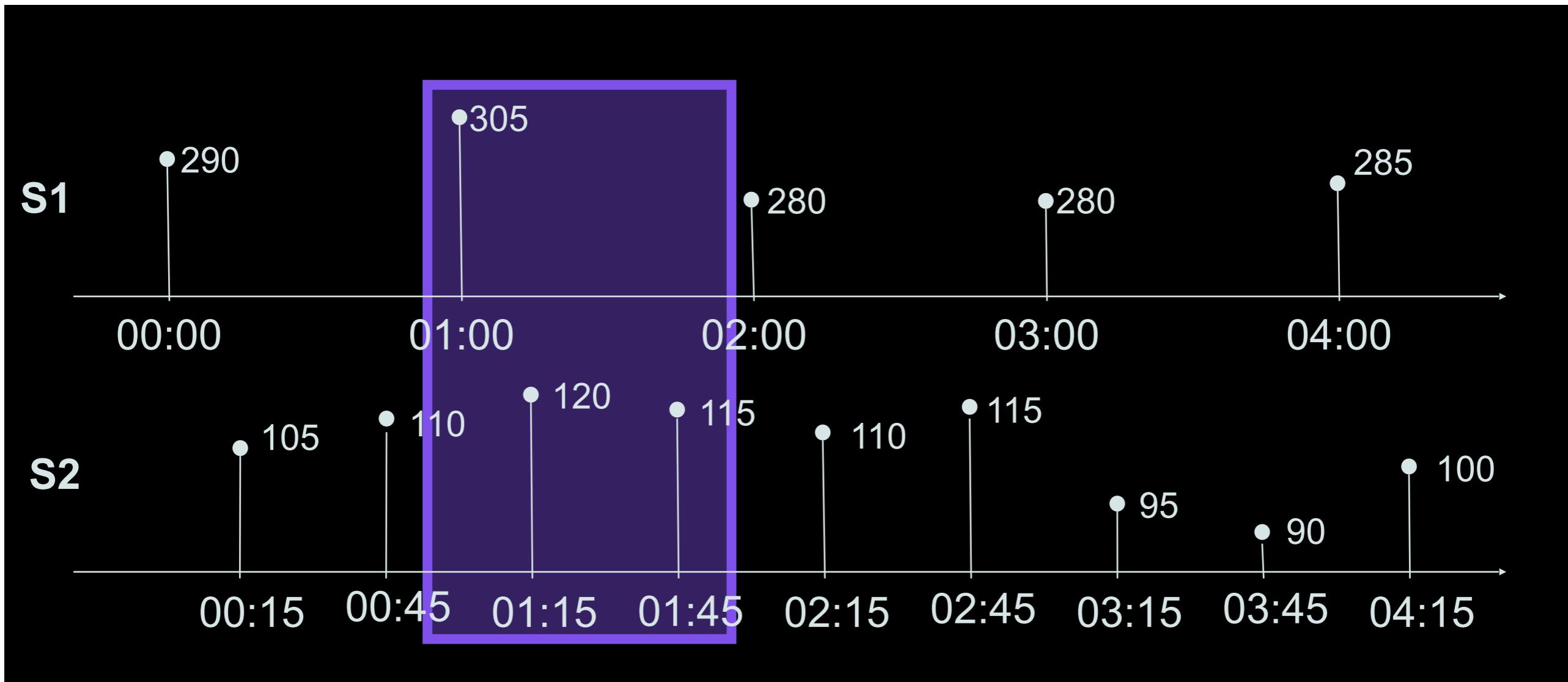
JOINING STREAMS IT'S OF SYNCRONIZATION



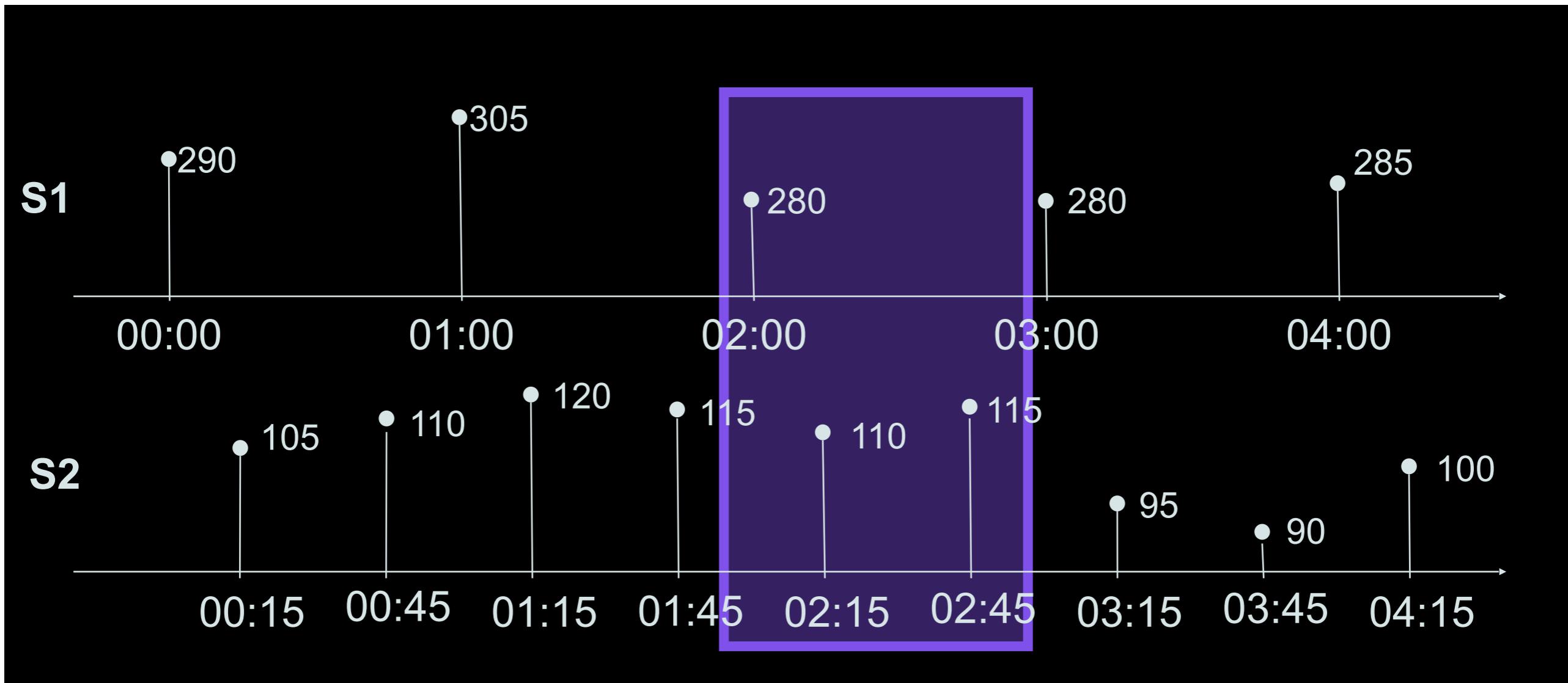
TUMBLING WINDOWS CAN SYNCHRONIZE STREAMS!



TUMBLING WINDOWS CAN SYNCHRONIZE STREAMS!



TUMBLING WINDOWS CAN SYNCHRONIZE STREAMS!



STREAM PROCESSING VS. REQUIREMENTS

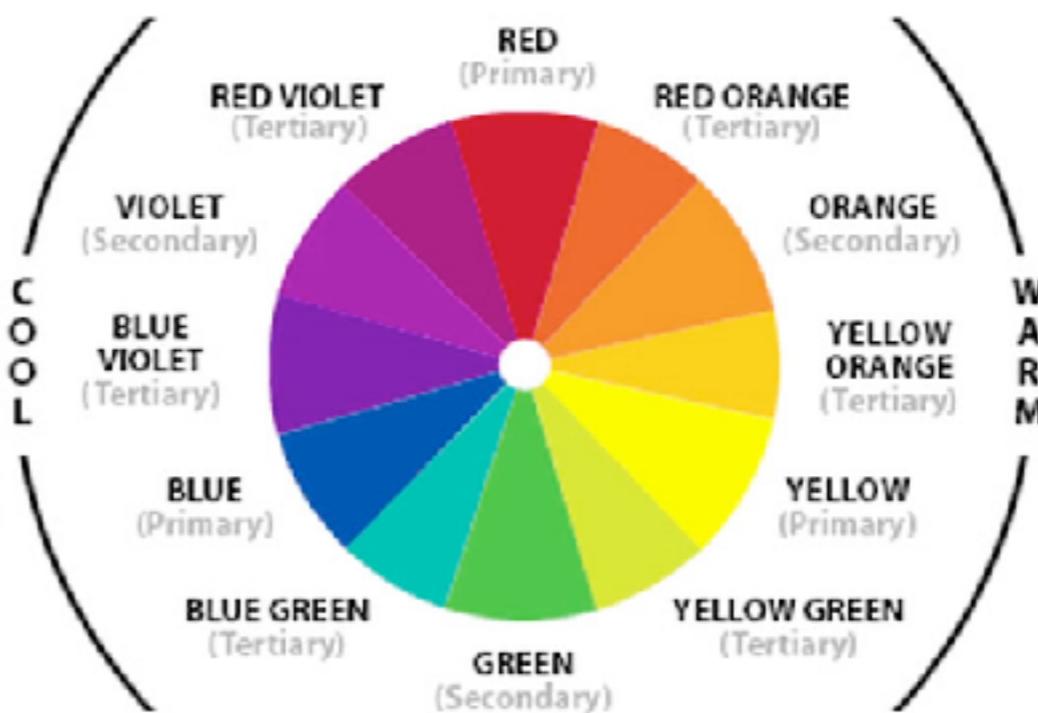
Requirement	SP
massive datasets	✓
data streams	✓
heterogeneous dataset	✗
incomplete data	✗
noisy data	✓
reactive answers	✓
fine-grained information access	✓
complex domain models	✗

SEMANTIC WEB TECHS A USER PERSPECTIVE

Are there any **cool** colored box?

yes, 7 , 13 , ...

An ontology of colors



1 minute wide window



time

DATA MODEL

▶ RDF: Resource Description Framework

- ▶ It allows to make statements about resources in the form of subject-predicate-object expressions

- ▶ In RDF terminology triples

▶ E.g. $\underbrace{:Alice}_{\text{subject}}$ $\underbrace{:posts}_{\text{predicate}}$ $\underbrace{"I'm \ with \ @Bob"}_{\text{object}}$



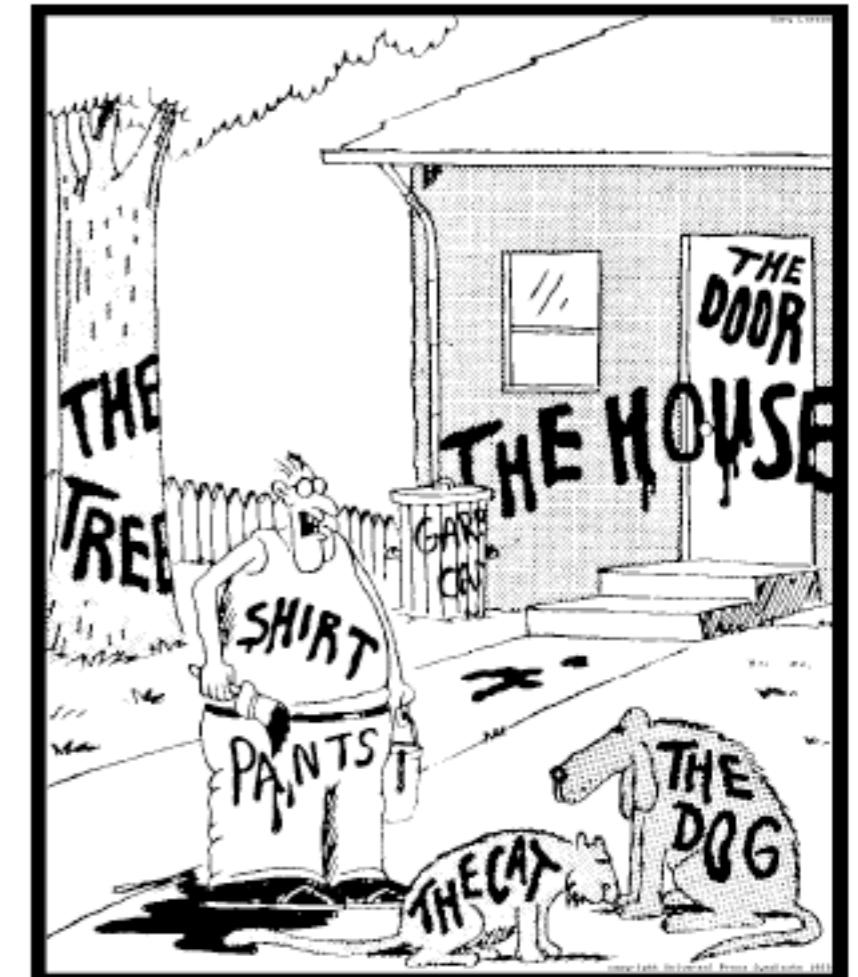
- ▶ A collection of RDF statements represents a labelled, directed graph

- ▶ In RDF terminology a graph

- ▶ E.g., the triple above can be connected to millions of others telling information about Rodin and The Thinker

ONTOLOGICAL LANGUAGE

- ▶ OWL: Web Ontology Language
 - ▶ It allows to give **well-defined meaning** to classes, properties, individuals, and data values
 - ▶ E.g.
 - ▶ posts **is a** property
 - ▶ posts **is a subproperty of** communicates
 - ▶ those who posts **are** social media users
 - ▶ ...
 - ▶ A collection of classes, properties, individuals, and data values forms a **vocabulary**
 - ▶ When using **OWL2DL**, **classes and properties** are isolated **in a T-box** while **individuals and values** are **in an A-box**



QUERY LANGUAGE

- ▶ **SPARQL**: Querying RDF under OWL entailment regime

- ▶ It allows to make search for **statements about resources**

- ▶ In SPARQL terminology a **triples pattern** is an RDF triple in which users can add variables

- ▶ E.g. 1: what does Alice post? :Alice :posts ?x

- ▶ E.g. 2: what does Alice communicate? :Alice :communicates ?y

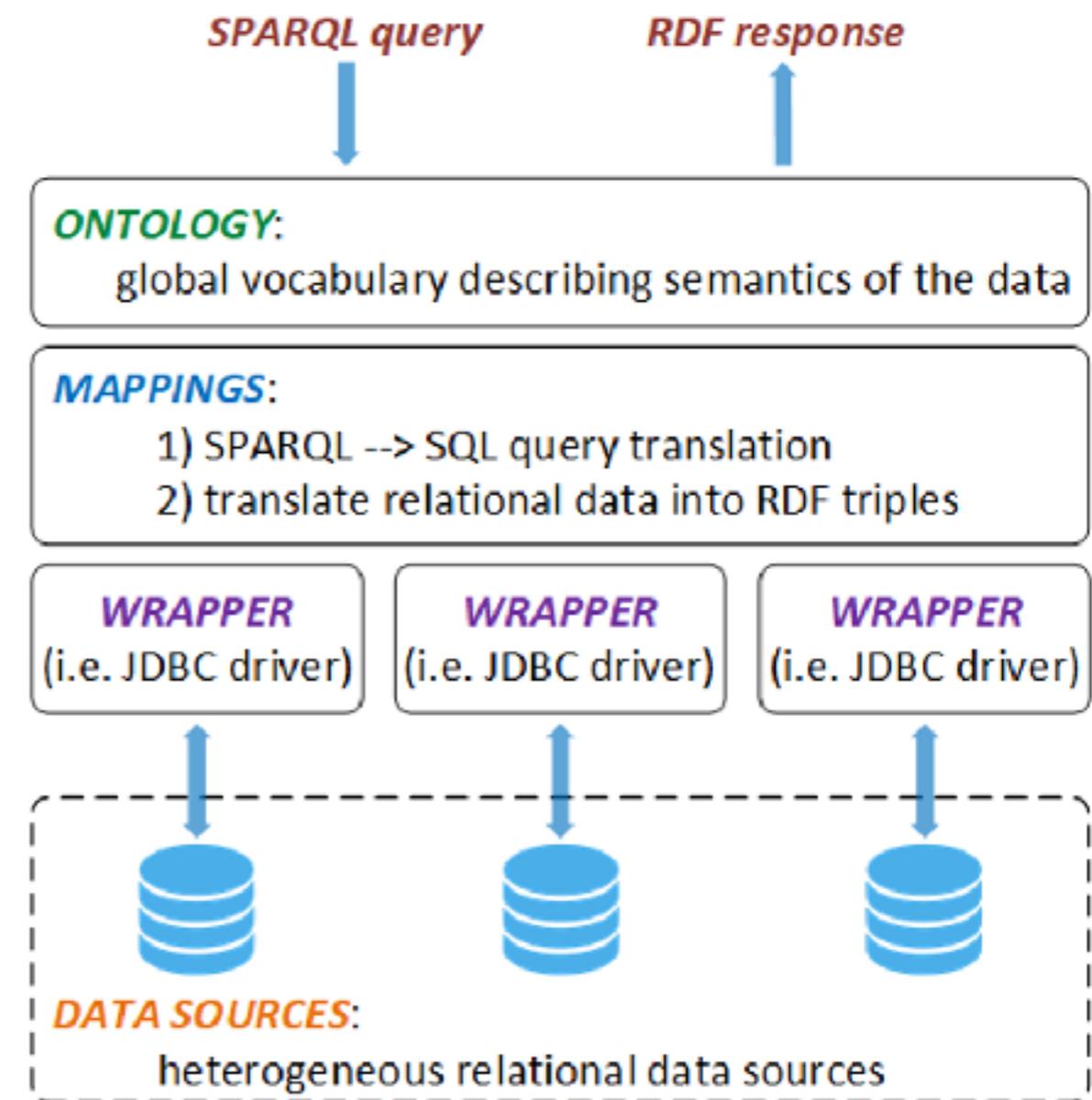
- ▶ E.g. 3: Is there a Social Media User? ?x a :SocialMediaUser?

- ▶ A collection of triples patterns represents a **graph pattern**

- ▶ Graph patterns can be combined with FILTER, UNION and other clauses to form an expressive query language

ONTOLOGY-BASED DATA ACCESS

- ▶ Accessing data in SQL DB via SPARQL + OWL2QL
 - ▶ queries expressed using terms whose semantics is specified in an ontology
 - ▶ one or more SQL DB appear as a single RDF repository a.k.a. a **Virtual Knowledge Graph**



ONTOLOGY-BASED DATA ACCESS - A SIMPLE EXAMPLE

- ▶ Original SPARQL query
 - ▶ what does Alice communicate?
 - ▶ :Alice :communicates ?y
- ▶ Knowing that posts is a subproperty of communicates we know
 - ▶ :Alice :posts ?y implies :Alice :communicates ?y
 - ▶ If triples such as.
are mapped to the results of
`: {USER} :posts {TEXT}`
`SELECT USER, TEXT FROM POSTS`
 - ▶ we can rewrite
in SQL as
`:Alice :posts ?y`
`SELECT USER, TEXT FROM POSTS`
`WHERE USER="Alice"`
- ▶ the SQL answers are *certain answers* of the original SPARQL query

SEMANTIC WEB TECHS VS. REQUIREMENTS

Requirement	SP	ST
massive datasets	✓	✓
data streams	✓	✗
heterogeneous dataset	✗	✓
incomplete data	✗	✓
noisy data	✓	✗
reactive answers	✓	✗
fine-grained information access	✓	✓
complex domain models	✗	✓

WEB STREAM PROCESSING RESEARCH QUESTION

Is it possible to **process** in **real time**, **multiple**,
heterogeneous, **gigantic** and **inevitably noisy**
and **incomplete** **Web streams** ?

R. Tommasini, P. Bonte, E. Della Valle, 2022

PRELIMINARY SETUP

- ▶ During this introduction, you can get ready for the hands-on sessions:
- ▶ Install docker and docker-compose:
 - ▶ docker: <https://docs.docker.com/engine/install/>
 - ▶ docker-compose: <https://docs.docker.com/compose/install/>
- ▶ Clone our repository:
<https://github.com/pbonete/WSP-TheWebConf2022Tutorial>
 - ▶ e.g., `git clone https://github.com/pbonete/WSP-TheWebConf2022Tutorial.git`
 - ▶ Also downloading the zip is ok

GETTING READY FOR THE FIRST HANDS-ON

- ▶ Downloading the docker images make take a while ...
- ▶ You'd better start the platform for the first hands-on:
 - ▶ Open a terminal
 - ▶ Go the right folder: `cd exercises/part1`
 - ▶ Bring up the stack: `docker-compose up`

WEB STREAM PROCESSING A USER PERSPECTIVE

Which are the top-2 most frequent **cool** colors in the last minute?

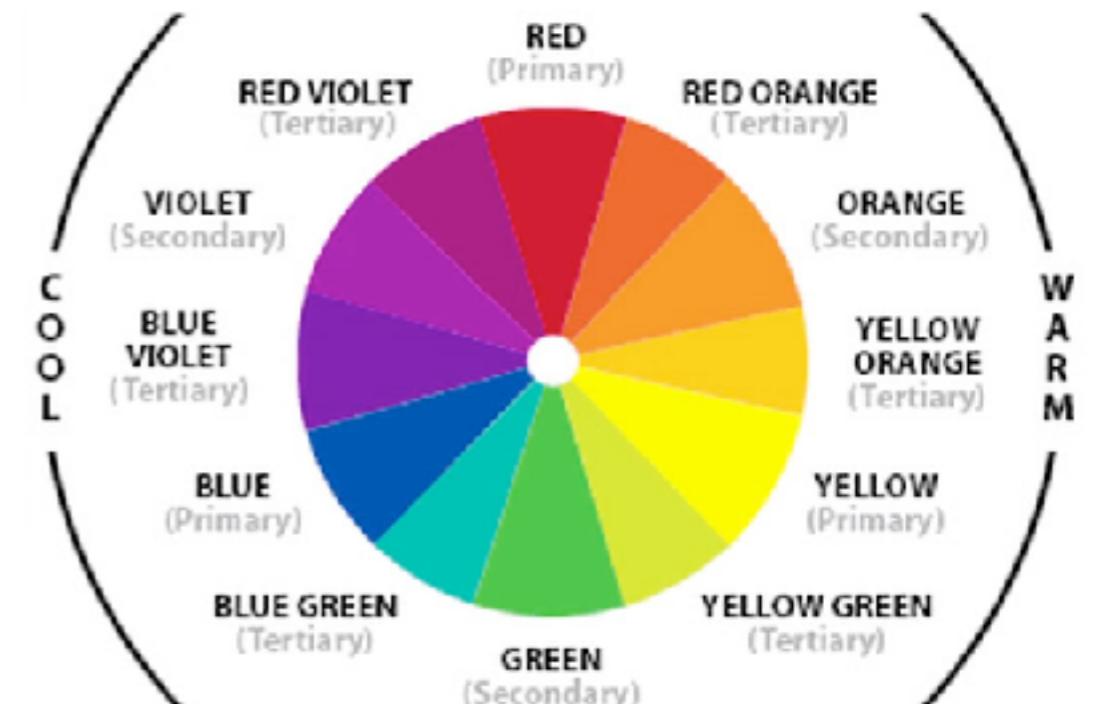
(, 13), (, 8), (, 8)

1 minute wide window



time

An ontology of colors



DATA MODEL

- ▶ **RDF STREAM:** Stream of Resource Description Framework
 - ▶ It allows to make **timestamped statements about resources** in the form of **subject-predicate-object expressions**
 - ▶ In RDF terminology **triples**
 - ▶ E.g.

:Alice :posts "I'm with @Bob" "2022-4-26"

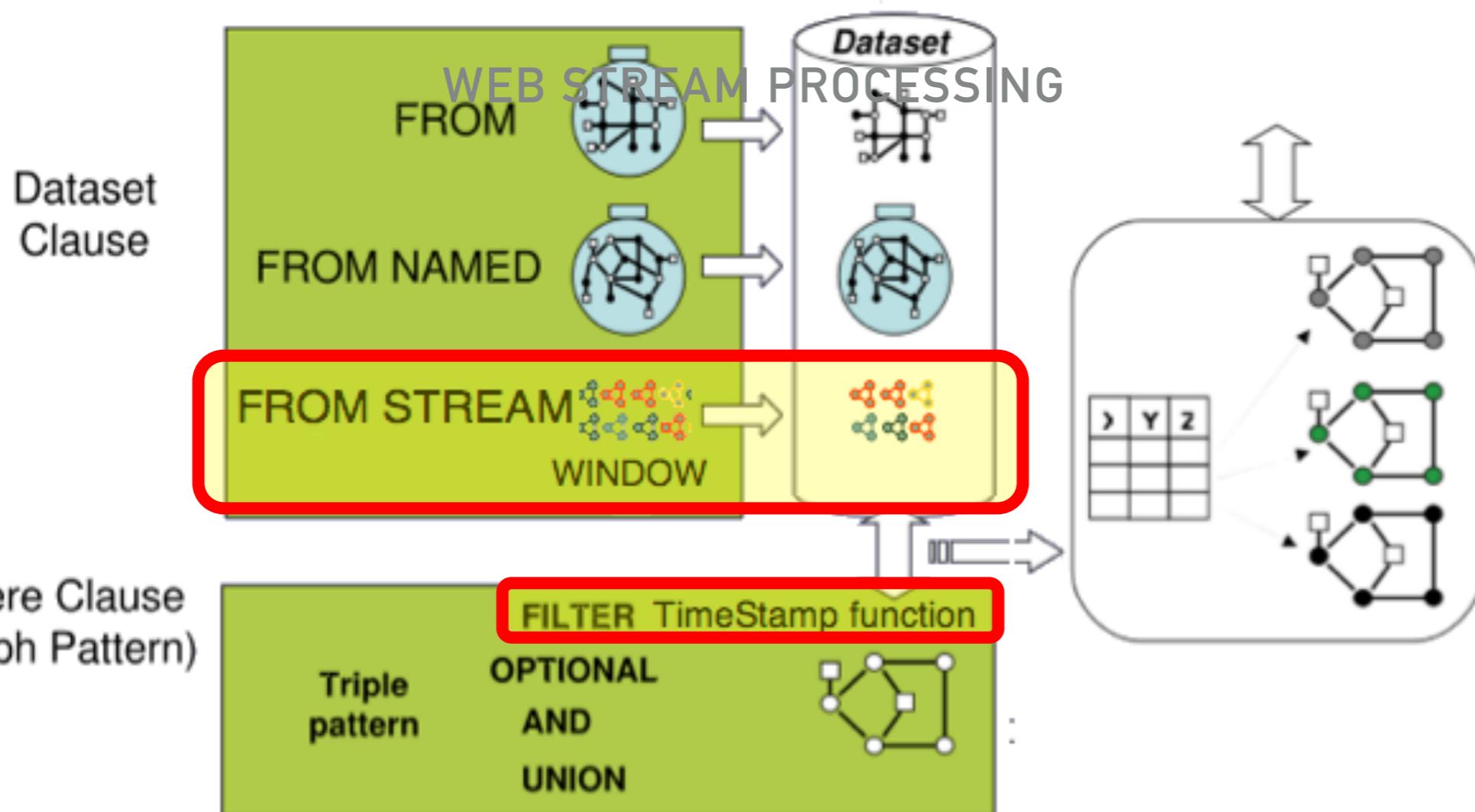
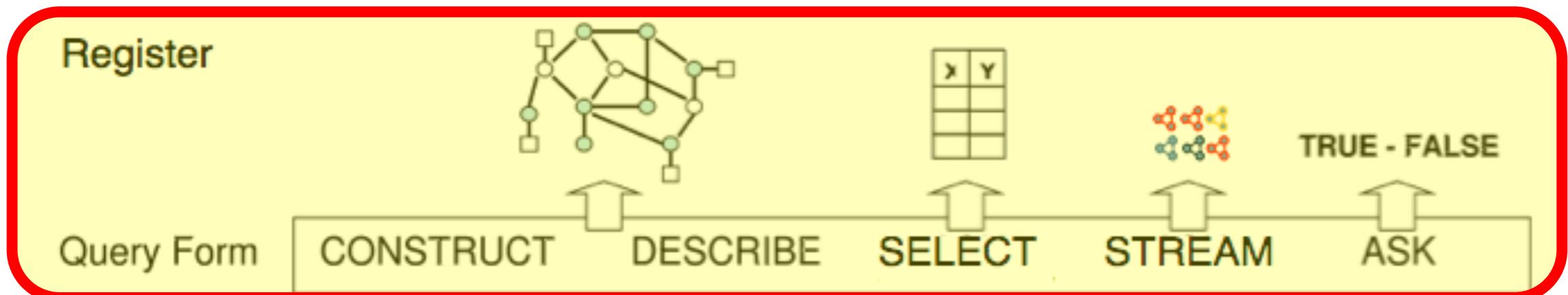
subject predicate object timestamp

The diagram illustrates an RDF triple. It consists of four parts: a subject (:Alice), a predicate (:posts), an object ("I'm with @Bob"), and a timestamp ("2022-4-26"). Brackets below the subject and predicate are labeled "subject" and "predicate" respectively. Brackets below the object and timestamp are labeled "object" and "timestamp" respectively.
 - ▶ An **unbounded sequence of timestamped RDF statements** represents an **RDF stream**

CONTINUOUS QUERY LANGUAGE

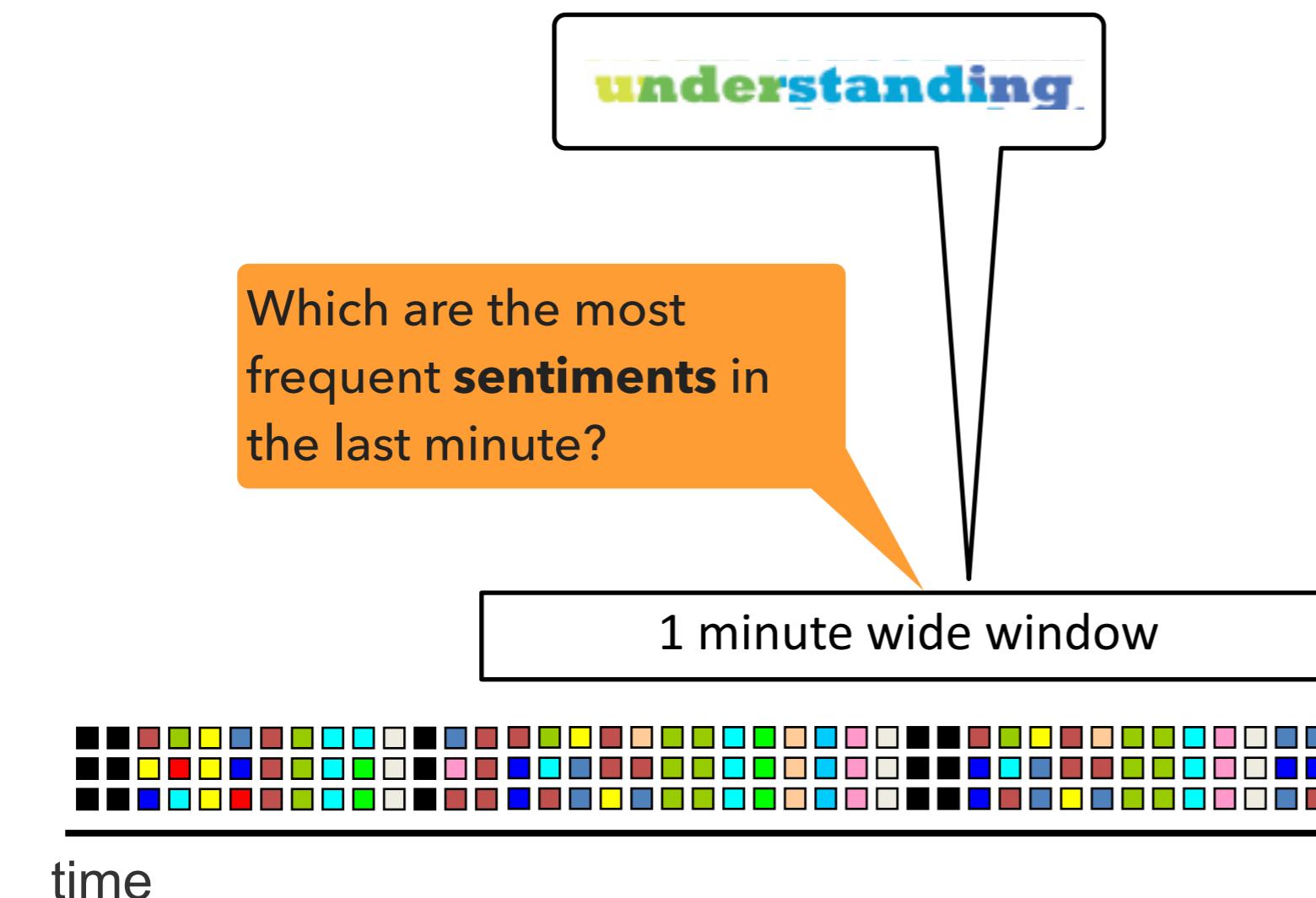
- ▶ RSP-QL: RDF stream processing query language
- ▶ It allows to register continuous queries that process RDF streams
 - ▶ E.g. who have been the most active users in the last minute?
 - ▶ **REGISTER RSTREAM** <MostActiveUsers> AS
SELECT ?user COUNT (?x)
FROM NAMED WINDOW <w1>
ON <SocialMediaStream> [**RANGE PT1M STEP PT1M**]
WHERE { WINDOW <w1> { ?user :posts ?x } }
GROUP BY ?user

HOW RQLQL EXTENDS SPARQL TO PROCESS RDF STREAMS



WEB STREAM PROCESSING A USER PERSPECTIVE

The **better** is the **ontology** (of the colors) we are using
the **more expressive** are the **queries** we can register



A better
ontology (of colors)

b a l a n c e
warmth vibrant ex-
pansive demand attention
controversy flamboyant
energy activity appetite social-
ization blood heat vigor passionate
intense fierce love danger exciting
strength irritating lips hearts sexy ro-
mance sensuality impulsive leadership
courage competence independence orga-
nization self-motivation spirituality plea-
sure vitality will to win survival instinct intu-
ition entrepreneurial desire fire stimulation
joy rage sunshine tropical enthusiasm fasci-
nation happiness creativity attraction success
citrus endurance illumination wisdom wealth
intellect loyalty freshness growth harmony fer-
tility safety money vision experience novice
hope nature finance ambition greed jealousy
healing protection peace sky sea depth trust
confidence faith truth heaven mind tranquil-
ity calm sincerity clean water mineral preci-
sion expertise understanding softness
knowledge power royalty nobility luxury
extravagance dignity mystery magic arti-
ficial nostalgia gloom frustration light
goodness innocence purity perfec-
tion positive beginning cool sim-
plicity charity angels sterility el-
egance formality evil fear
unknown feeling author-
ity prestige grief
h a r m o n y

HANDS ON TIME

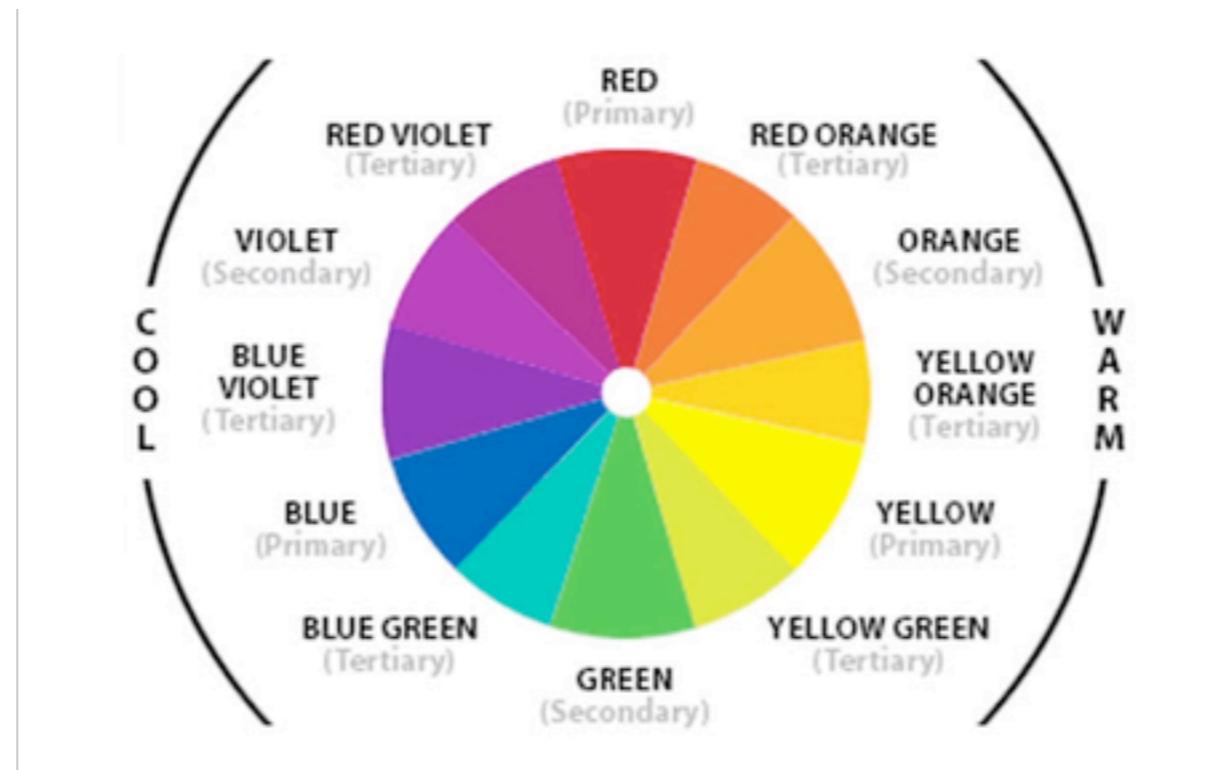
COLORWAVE!



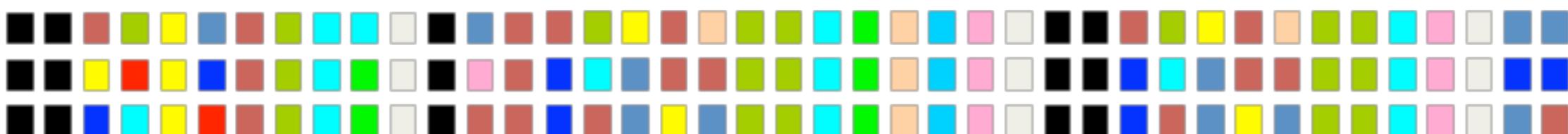
GETTING READY FOR THE FIRST HANDS-ON

- ▶ Start the platform for the first hands-on:
 - ▶ Open a terminal
 - ▶ Go the right folder: `cd exercises/part1`
 - ▶ Bring up the stack: `docker-compose up`

COMPARING WARM AND COLD COLORS

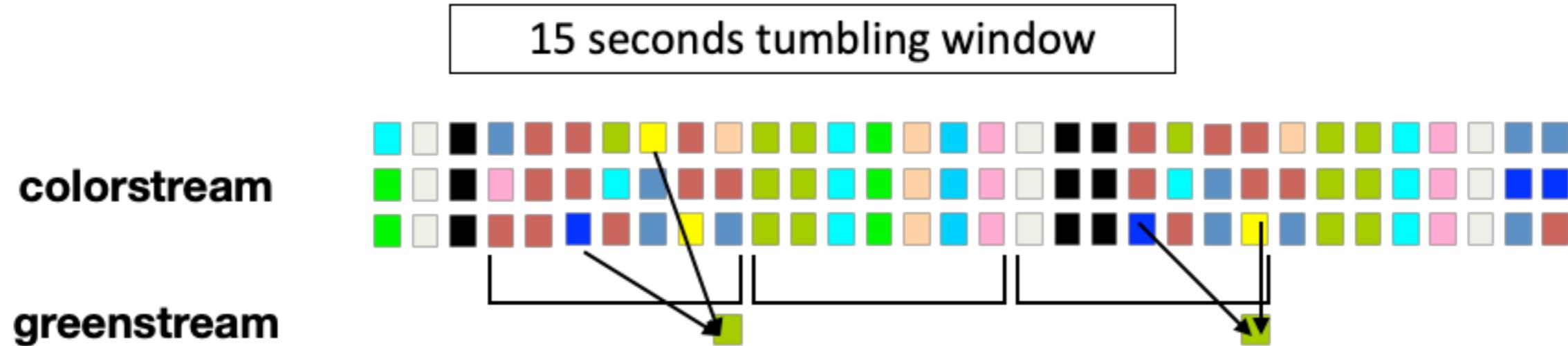


1 minute tumbling window



warmCount = 34
coldCount = 26

COOKING GREENS FROM YELLOWS AND BLUES





GHENT
UNIVERSITY

imec



POLITECNICO
MILANO 1863



QUANTIA
consulting

WEB STREAM PROCESSING

WITH RSP4J AND ONTOPSTREAM

TheWebConf 2022, Online, hosted by Lyon, France - 26-4-2022