Tutorial Paper

Streams Meet Semantics: Foundations and Systems of RDF Stream Processing

Haridimos Kondylakis
kondylak@ics.forth.gr
FORTH-ICS & Univ. of Crete
Greece

Pieter Bonte
pieter.bonte@kuleuven.be
Department of Computer Science, KU Leuven
Belgium

Olivier Curé
olivier.cure@univ-eiffel.fr
LIGM, CNRS - Gustave Eiffel University
France

Riccardo Tommasini
riccardo.tommasini@insa-lyon.fr
INSA Lyon, CNRS & Tartu University
France and Estonia

Abstract

The increasing volume and velocity of data generated by sources such as the Internet of Things, social media, and cyber-physical systems have intensified the need for real-time data processing solutions. At the same time, the Resource Description Framework (RDF) has become a central model for representing and integrating heterogeneous data on the Web. These trends have given rise to RDF Stream Processing (RSP), which combines stream processing with Semantic Web technologies to support continuous querying, temporal analysis, and reasoning over RDF data streams.

This tutorial provides a comprehensive overview of RDF Stream Processing, focusing on its query languages, systems, and benchmarks. We introduce the core concepts and challenges of RSP, present a structured taxonomy of existing languages and systems, and discuss their design choices, execution models, and reasoning capabilities. We also review benchmarking efforts used to evaluate RSP solutions and highlight open research challenges and emerging directions. The tutorial aims to equip researchers and practitioners with a clear understanding of the RSP landscape and guidance for selecting appropriate approaches for different application scenarios.

Keywords

Stream Reasoning, RDF, Semantics, Foundations, Languages, Systems

1 Introduction

The last decade has witnessed an unprecedented growth in the volume, velocity, and heterogeneity of data generated by modern applications. Internet of Things (IoT) deployments, social media platforms, smart cities, and cyber-physical systems continuously produce large streams of events that must be processed in near real time. At the same time, there is an increasing need to enrich such data with semantics in order to enable interoperability [8], high-level querying [11, 12], and reasoning. The Resource Description Framework (RDF), as the foundational data model of the Web of Data, has emerged as a natural choice for representing and integrating heterogeneous data sources using shared vocabularies and ontologies.

The combination of streaming data and RDF-based knowledge representation has led to the emergence of RDF Stream Processing (RSP), a research area situated at the intersection of data stream management systems, complex event processing, and Semantic Web technologies. RSP extends traditional stream processing by supporting rich graph-based data models, declarative query languages, temporal reasoning, and background knowledge integration. These capabilities enable advanced applications such as real-time monitoring, anomaly detection, and situation awareness over semantically annotated data streams.

However, processing RDF streams efficiently poses significant challenges. RDF streams are potentially unbounded, highly dynamic, and temporally annotated, requiring specialized data models and execution strategies. The need to combine continuous querying with reasoning over ontologies further complicates system design, raising issues related to scalability, incremental maintenance, temporal semantics, and correctness. Over the years, a wide variety of approaches have been proposed, including extensions of SPARQL with windowing and event operators, rule-based and logic-based languages, centralized and distributed RSP engines, and dedicated benchmarks for evaluating expressiveness and performance.

Despite this rich body of work, the RSP landscape remains fragmented. Different languages and systems adopt different assumptions regarding time models, execution semantics, reporting policies, and reasoning capabilities, making it difficult for researchers and practitioners to understand their relationships, trade-offs, and suitability for specific use cases. Furthermore, recent advances in distributed stream processing, edge computing, and large-scale data integration have introduced new design choices that are not always captured in earlier overviews.

This tutorial aims to address this gap by providing a systematic and up-to-date introduction to RDF Stream Processing, grounded in our comprehensive survey of the field, recently published in VLDBJ [4]. We organize existing work along clear conceptual dimensions, covering languages, systems, and benchmarks, and highlight both foundational principles and recent developments. By doing so, the tutorial seeks to offer a coherent view of the RSP design space, facilitate informed technology choices, and stimulate further research at the intersection of streaming data management and semantic technologies.

In recent years, several tutorials at major database conferences such as SIGMOD, EDBT, VLDB, and ICDE have addressed aspects of data stream processing [5], languages [2], performance tuning [7], and machine learning pipelines [6]. These tutorials typically focus on general-purpose stream processing architectures, SQL-based or declarative streaming languages, or streaming analytics and machine learning. Other tutorials, mainly outside the core database venues, have considered stream reasoning using specific tools, engines, or semantic Web technologies in isolation

EDBT '26, Tampere (Finland)

10.48786/edbt.2026.80

[1, 3, 10]. In contrast, no recent tutorial at these venues has provided a comprehensive and unified treatment of RDF Stream Processing that jointly covers its foundational concepts, query languages, execution systems, and benchmarking methodologies. The proposed tutorial fills this gap by offering an end-to-end view of the RSP landscape, grounded in a recent comprehensive survey, and by explicitly framing RDF stream processing challenges from a data management perspective. By systematically comparing languages, systems, and evaluation approaches, and by highlighting their design trade-offs and open challenges, this tutorial complements existing offerings and provides novel, non-overlapping value to the EDBT/ICDT community.

Tutorial Material. The tutorial will use slides with code samples, datasets, and best practices. All content will be available on the tutorial web page before March under a CC license. *No additional audiovisual or technical equipment is required.*

2 Tutorial Structure and Duration.

This tutorial is 2 hours long and divided into four parts.

Part I (Section 5.1) (20 mins), presented by Oliver Curé, introduces the background and foundations of RDF Stream Processing. We present the motivation and application context of RDF streams, review the RDF data model and streaming abstractions, and introduce key temporal concepts such as windows, reporting policies, and stream semantics. This part establishes the necessary background for understanding language and system design choices.

Part II (Section 5.2) (30 mins), presented by Oliver Curé, focuses on RDF Stream Processing languages. We survey SPARQL-based and rule-based approaches, discussing their design principles, temporal models, and formal semantics. The discussion highlights common abstractions, expressiveness trade-offs, and differences in reporting and reasoning capabilities across languages.

Part III (Section 5.3 and Section 5.4) (40 mins), presented by Riccardo Tommasini, provides an in-depth exploration of RDF Stream Processing systems. We review representative RSP systems, covering centralized, distributed, and federated architectures, and analyze their execution models, scalability, and reasoning support. RSP4J [9] open-source library will serve as a reference for providing a unified software solution to the attendant. We then discuss benchmarking and evaluation methodologies, including datasets, query workloads, and performance metrics used to assess RSP solutions.

Part IV (Section 5.5) (30 mins), presented by Haridimos Kondylakis, is devoted to open challenges and future research directions in RDF Stream Processing. Topics include unifying languages, semantics, and systems; scalable, incremental reasoning over high-velocity streams; support for distributed and edge-based deployments; interoperability; and the integration of RSP with large-scale knowledge graphs, machine learning pipelines, and AI-driven techniques.

The tutorial is organized into two sessions. The first session covers Parts I and II, allowing participants to build a solid understanding of the foundations and language-level aspects of RDF Stream Processing. The second session covers Parts III and IV, focusing on systems, benchmarking, and a forward-looking discussion of open challenges and future directions. redIn particular, we will also expand on newer graph stream processing techniques that emerged after the publication of the initial survey.

3 Tutorial Relevance and Goals

RDF Stream Processing (RSP) has become an important research topic at the intersection of data management, stream processing, and Semantic Web technologies. The increasing adoption of RDF and knowledge graphs in dynamic environments such as IoT, smart cities, social media analytics, and real-time monitoring systems has highlighted the need for expressive and scalable solutions that can process semantically annotated data streams in a timely manner.

The EDBT/ICDT community has a long-standing interest in stream processing, continuous query evaluation, temporal data management, and reasoning over structured data. RSP naturally builds upon these foundations by combining classical stream processing concepts with graph-based data models and declarative query languages. In recent years, a significant number of RSP languages, systems, and benchmarks have been proposed, addressing challenges such as window semantics, incremental evaluation, reasoning under streaming updates, and scalability across distributed and edge environments.

The primary goal of this tutorial is to provide a comprehensive and structured overview of the RSP landscape. Specifically, the tutorial aims to:

- introduce the core concepts and challenges underlying RDF Stream Processing;
- present a systematic taxonomy of RSP query languages, highlighting their design principles, temporal abstractions, and formal semantics;
- survey existing RSP systems, analyzing their architectures, execution models, and reasoning capabilities;
- review benchmarking efforts and evaluation methodologies used to assess RSP solutions;
- discuss open challenges and emerging research directions in the field.

By synthesizing more than a decade of research results and recent advances, the tutorial is designed to help attendees understand the trade-offs between different approaches and select appropriate techniques for their specific application needs.

4 Target Audience and Prerequisites

The tutorial is intended for a broad audience within the EDBT/ICDT community, including:

- researchers interested in data stream management, temporal data processing, and knowledge graphs;
- practitioners and system developers working on real-time analytics over semantic data;
- PhD students and early-career researchers seeking an entry point into RDF Stream Processing.

The tutorial is largely self-contained. A basic understanding of database systems and query processing is assumed. Familiarity with RDF and SPARQL is helpful but not strictly required, as the necessary background will be introduced during the tutorial.

5 Tutorial content

5.1 Background and Foundations

This section introduces the foundational concepts required to understand RDF Stream Processing. We first review the RDF data model and its role in representing semantically enriched data. We then introduce the notion of data streams, emphasizing the challenges posed by unbounded, time-ordered data.

We discuss how RDF streams extend the classical RDF model by associating timestamps with RDF triples or graphs, and we introduce common temporal abstractions such as time-based and tuple-based windows. We also briefly review background technologies from the stream processing ecosystem, including data stream management systems and complex event processing, and explain how they influence the design of RSP languages and systems.

5.2 RDF Stream Processing Languages

This section provides a structured overview of query languages proposed for RDF Stream Processing. We organize existing languages along several key dimensions, including syntax, design patterns, temporal models, and formal semantics. The dimensions of study for the various languages is shown in Figure 1.

We first present SPARQL-based approaches, which extend the W3C standard query language with constructs for windowing, continuous querying, and event detection. These languages typically preserve SPARQL's graph pattern matching semantics while introducing stream-specific operators.

We then discuss rule-based approaches, including languages grounded in Datalog, Answer Set Programming, and Description Logics. These languages often emphasize reasoning and complex event recognition, supporting expressive temporal patterns and non-monotonic semantics.

For each category, we analyze the strengths and limitations of representative languages, focusing on expressiveness, ease of use, and suitability for different application scenarios.

5.3 RDF Stream Processing Systems

This section surveys systems that implement RDF Stream Processing languages and execution models. We classify systems according to their architecture (centralized, distributed, or federated) and discuss their ingestion mechanisms, query evaluation strategies, and reasoning support. An overview of the various dimensions of study is shown in Figure 2.

We analyze centralized engines that emphasize expressiveness and incremental reasoning, as well as distributed, scalable systems built on modern stream processing frameworks. We also discuss emerging systems targeting fog and edge environments.

Throughout the section, we highlight design trade-offs across scalability, latency, fault tolerance, and reasoning capabilities, and discuss how these trade-offs influence system behavior in practice.

5.4 Benchmarking and Evaluation

To assess the performance and expressiveness of RSP solutions, several benchmarks have been proposed. This section reviews the main benchmarking efforts in the field, describing their datasets, query workloads, and evaluation metrics.

We discuss how benchmarks evaluate different aspects of RSP systems, including query coverage, throughput, latency, correctness, and scalability. We also analyze the limitations of existing benchmarks and identify open challenges in designing comprehensive and representative evaluation methodologies for RDF Stream Processing.

5.5 Open Challenges and Future Directions

Despite significant progress, RDF Stream Processing remains an active research area with many open challenges. In this section, we discuss issues such as:

- unifying semantics across languages and systems;
- scalable and incremental reasoning over high-velocity streams;
- support for distributed, federated, and edge-based deployments;
- integration with emerging data management paradigms, including large-scale knowledge graphs and machine learning pipelines.

We conclude by outlining promising research directions that can further advance the state of the art in RDF Stream Processing.

6 Conclusion

This tutorial provides a comprehensive overview of RDF Stream Processing, covering its theoretical foundations, practical systems, and evaluation methodologies. By organizing existing work along clear conceptual dimensions, the tutorial aims to clarify the design space of RSP and support informed decision-making for both researchers and practitioners.

We believe that RDF Stream Processing will continue to play a crucial role in enabling real-time, semantics-aware data management, and we hope that this tutorial will stimulate further research and collaboration within the EDBT/ICDT community.

7 Presenters

The tutorial will be presented by researchers with extensive experience in RDF Stream Processing, Semantic Web technologies, and data management systems. The presenters have contributed to foundational research in the area, including the design and analysis of RSP languages and systems, and have published extensively in leading database and Semantic Web venues.

Haridimos Kondylakis is an Associate Professor of Big Data Engineering in the Department of Computer Science, University of Crete, and an Affiliated Researcher at FORTH-ICS. His interests include Big Data Management, Semantics, Stream Processing and their application in Health. He has delivered multiple high-profile tutorials (e.g., EDBT, ISWC, VLDB) on related topics.

Pieter Bonte is an Assistant Professor in the Department of Computer Science at KU Leuven. His research focuses on RDF Stream Processing, stream reasoning, and Semantic Web technologies. He has co-authored several key publications and surveys in the area and has been actively involved in the design and analysis of languages and systems for processing semantic data streams.

Olivier Curé is a Professor at Université Gustave Eiffel. His research interests include Semantic Web technologies, knowledge graphs, data integration, and stream reasoning. He has contributed extensively to the development of RDF stream processing languages and systems and has significant experience in presenting tutorials and lectures on Semantic Web and data management topics at international venues (e.g., IEEE Big Data, WWW, ISWC).

Riccardo Tommasini is an Associate Professor at INSA Lyon and Visiting Professor at Tartu Univeristy. His research interests cover continuous queries, streaming graph processing, and programming languages. He has presented many tutorials on the above topics, e.g., VLDB'25, SIGMOD'24, ISWC'24, RW'19.

Acknowledgments. Tommasini is supported by the French Research Agency under grant agreement nr. ANR-22-CE23-0001 Polyflow. Haridimos Kondylakis was supported by the MULTIPULM EU HORIZON project (Grant Agreement 101226783).

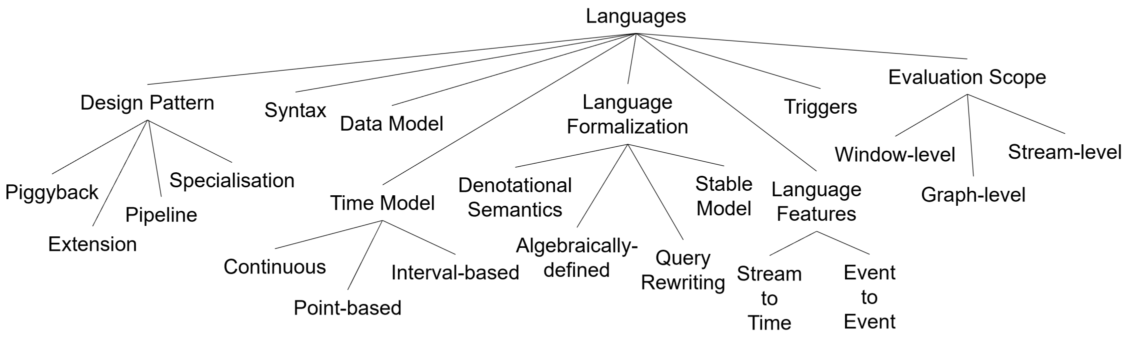

Figure 1: Dimensions of study for the various languages (adapted from [4]).

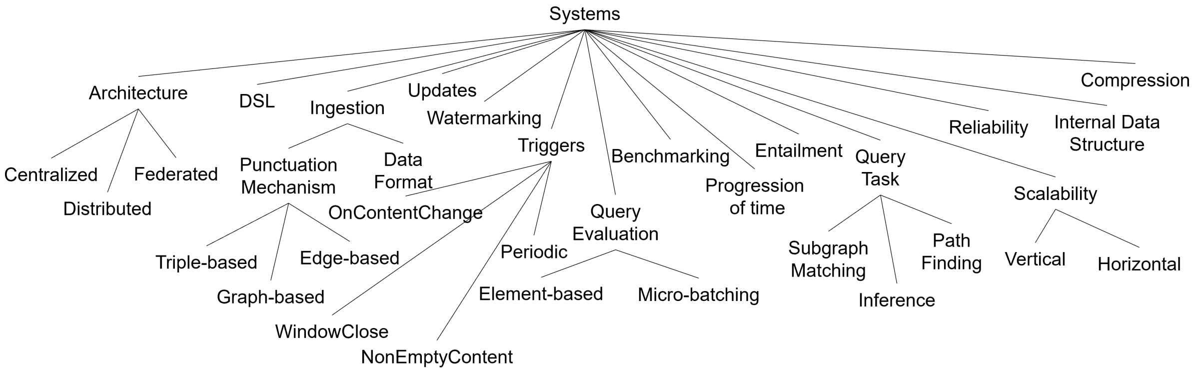

Figure 2: Dimensions of study for the various systems (adapted from [4]).

References

[1] 2026. Stream Reasoning. https://streamreasoning.org/events/. Accessed: 2026-01-19.

[2] Angela Bonifati and Riccardo Tommasini. 2024. An Overview of Continuous Querying in (Modern) Data Systems. In *Companion of the 2024 International Conference on Management of Data, SIGMOD/PODS 2024, Santiago, Chile, June 9-15, 2024*, Pablo Barceló, Nayat Sánchez-Pi, Alexandra Meliou, and S. Sudarshan (Eds.). ACM, 605–612. doi:10.1145/3626246.3654679

[3] Pieter Bonte, Jean-Paul Calbimonte, Daniel de Leng, Daniele Dell'Aglio, Emanuele Della Valle, Thomas Eiter, Federico Giannini, Fredrik Heintz, Konstantin Schekotihin, Danh Le Phuoc, Alessandra Mileo, Patrik Schneider, Riccardo Tommasini, Jacopo Urbani, and Giacomo Ziffer. 2024. Grounding Stream Reasoning Research. *TGDK* 2, 1 (2024), 2:1–2:47. doi:10.4230/TGDK.2.1.2

[4] Pieter Bonte, Christophe Callé, Olivier Curé, Haridimos Kondylakis, and Riccardo Tommasini. 2025. Languages and systems for RDF stream processing, a survey. *VLDB J.* 34, 4 (2025), 50. doi:10.1007/S00778-025-00927-7

[5] Paris Carbone, Marios Fragkoulis, Vasiliki Kalavri, and Asterios Katsifodimos. 2020. Beyond Analytics: The Evolution of Stream Processing Systems. In *Proceedings of the 2020 International Conference on Management of Data, SIGMOD Conference 2020, online conference [Portland, OR, USA], June 14-19, 2020*, David Maier, Rachel Pottinger, AnHai Doan, Wang-Chiew Tan, Abdussalam Alawini, and Hung Q. Ngo (Eds.). ACM, 2651–2658. doi:10.1145/3318464.3383131

[6] Heitor Murilo Gomes, Nuwan Gunasekara, and Yibin Sun. 2025. Machine Learning on the Fly: A Hands-On Tutorial for Streaming Data. In *41st IEEE International Conference on Data Engineering, ICDE 2025, Hong Kong, May 19-23, 2025.* IEEE, 4513–4516. doi:10.1109/ICDE65448.2025.00342

[7] Herodotos Herodotou, Lambros Odysseos, Yuxing Chen, and Jiaheng Lu. 2022. Automatic Performance Tuning for Distributed Data Stream Processing Systems. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022.* IEEE, 3194–3197. doi:10.1109/ICDE53745.2022.00296

[8] Sophia Sideri, Vasilis Efthymiou, Dimitris Plexousakis, and Haridimos Kondylakis. [n. d.]. PG2RDF: Schema-Guided Transformation of Property Graphs to RDF. ([n. d.]).

[9] Riccardo Tommasini, Pieter Bonte, Femke Ongenae, and Emanuele Della Valle. 2021. RSP4J: An API for RDF Stream Processing. In *The Semantic Web - 18th International Conference, ESWC 2021, Virtual Event, June 6-10, 2021, Proceedings (Lecture Notes in Computer Science, Vol. 12731)*, Ruben Verborgh, Katja Hose, Heiko Paulheim, Pierre-Antoine Champin, Maria Maleshkova, Óscar Corcho, Petar Ristoski, and Mehwish Alam (Eds.). Springer, 565–581. doi:10.1007/978-3-030-77385-4_34

[10] Riccardo Tommasini, Pieter Bonte, Fabiano Spiga, and Emanuele Della Valle. 2023. *Streaming Linked Data.* Springer.

[11] Georgia Troullinou, Giannis Agathangelos, Haridimos Kondylakis, Kostas Stefanidis, and Dimitris Plexousakis. 2024. DIAERESIS: RDF data partitioning and query processing on SPARK. *Semantic Web* 15, 5 (2024), 1763–1789. doi:10.3233/SW-243554

[12] Giannis Vassiliou and Haridimos Kondylakis. 2026. Love-at-First-Sight: First Answers Without the Awkward Silence in Big Knowledge Graphs. In *VLDB.*