
CHALMERS
UNIVERSITY OF TECHNOLOGY

# V-Mon: Scalable and Fault-Tolerant Stream Processing Pipeline for Monitoring Vehicular Data Validity

# V-Mon: Scalable and Fault-Tolerant Stream Processing Pipeline for Monitoring Vehicular Data Validity

Carl-Magnus Wall
Volvo Group Trucks Technology
Gothenburg, Sweden
carl.magnus.wall@consultant.volvo.com

Måns Josefsson
Chalmers University of Technology
Gothenburg, Sweden
mans.josefsson99@gmail.com

Martin Hilgendorf
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden
martin.hilgendorf@chalmers.se

Marina Papatriantafilou
Chalmers University of Technology
and University of Gothenburg
Gothenburg, Sweden
ptrianta@chalmers.se

Binay Mishra
Volvo Group Trucks Technology
Gothenburg, Sweden
binay.mishra@volvo.com

## Abstract

With constant increases in edge devices in industry settings, increases in data rates naturally follow. However, with high, unbounded data rates, traditional (store-then-process) database procedures and batch-based processing are struggling to remain performant. To this end, processing streams of data continuously is an increasingly appealing approach, targeting low latency, high scalability and real-time data processing. This work examines design considerations as well as performance trade-offs for a stream processing pipeline targeting stateful analysis. The pipeline implementation employs Apache Kafka, Apache Flink and Apache Druid, and is studied through an example use case at Volvo Trucks, focusing on signal data set validity analysis. Performance evaluation of the pipeline reveals that the throughput requirements of the use case are satisfied, while also achieving sub-second latencies and offering a degree of fault tolerance. The pipeline also shows promise of adapting well to different levels of scale, providing enough headroom for a tenfold increase in data volumes over current demands. Further, the extensible nature of the pipeline enables the support of various feature extraction methods, e.g., data synopsis and sketching, and alternative data representations, e.g., knowledge graphs.

## CCS Concepts

• **Information systems** → **Data streaming**; • **General and reference** → **Design**; **Validation**; • **Computer systems organization** → **Pipeline computing**; *Redundancy*.

## Keywords

Data pipelines, stream processing, data validation, scalability, fault tolerance, latency & throughput, data completeness


*DEBS '25, Gothenburg, Sweden*

ACM ISBN 979-8-4007-1332-3/25/06
https://doi.org/10.1145/3701717.3733228

**ACM Reference Format:**
Carl-Magnus Wall, Måns Josefsson, Martin Hilgendorf, Marina Papatriantafilou, and Binay Mishra. 2025. V-Mon: Scalable and Fault-Tolerant Stream Processing Pipeline for Monitoring Vehicular Data Validity. In *The 19th ACM International Conference on Distributed and Event-based Systems (DEBS '25), June 10–13, 2025, Gothenburg, Sweden.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3701717.3733228

## 1 Motivation and overview

While larger volumes of data in industry deployments can be useful for analyses and decision-making, the process of transmitting, storing and extracting value from the data keeps introducing new challenges [2]. To accommodate large data flows from edge devices, and provide end users such as data analysts with timely, high-quality data, efficient and scalable *data pipelines* are a necessity [3].

As an alternative approach to traditional store-then-process data handling methods, *stream processing engines (SPEs)* are able to handle data in a continuous manner, at low latency and with high transformation granularity [4]. These are desirable benefits within data analytics since they provide quick responses in suitable data representations [2]. Pertaining to data analytics, one key factor is that of *data validity* [6]. Valid data helps to provide accurate, high-quality results from various analyses. Nevertheless, with high data volumes, low-latency data validation becomes cumbersome.

Through stream processing pipelines, data validation can be performed efficiently [6]; note though that such pipelines also introduce design considerations, including how to implement *parallelism and fault tolerance* for scalable, efficient and reliable operation. This work studies such considerations through a use case at Volvo Trucks [5]. Here, vehicular data of up to one billion data items are collected daily. A common validation and statistics process for this data includes *Not-a-Number (NaN)* value monitoring. The higher the NaN frequency, the lower the validity of the analyzed data set. Due to rising sensor density and data rates, this non-automated process has to discard data in certain scenarios. To mitigate this and similar problems, this work studies *stateful processing in data pipelining*, with the aforementioned simple, yet representative example as a use case, providing a proof-of-concept implementation, called V-Mon. The processing involves performing the validation and statistics maintenance tasks within time windows; the work

includes analysis of the design space and detailed study of the pipeline's properties. V-Mon features parallelism to enhance *efficiency*. To minimize loss risks, data replication is applied for *fault tolerance*, and its impact on pipeline performance is analyzed.

For the data validity monitoring, three pipeline layers are defined in V-Mon. First, a vehicle data generator, which creates sensor data payloads with a configurable NaN value rate. Second, a continuous data messaging and processing layer, which utilizes a stateful operator to track NaN value frequency over time windows (e.g., of one minute each). Third, a data-serving layer for live monitoring. Figure 1 shows the proposed pipeline design, which intends to be as extensible as possible, to allow for various data representations and feature extraction methods. The design is to satisfy the following requirements (ordered by precedence):

***R1*** Ensuring data completeness and fault tolerance
***R2*** Sustaining high throughput, satisfying an expected daily rate
***R3*** Achieving sub-second latencies and high scalability

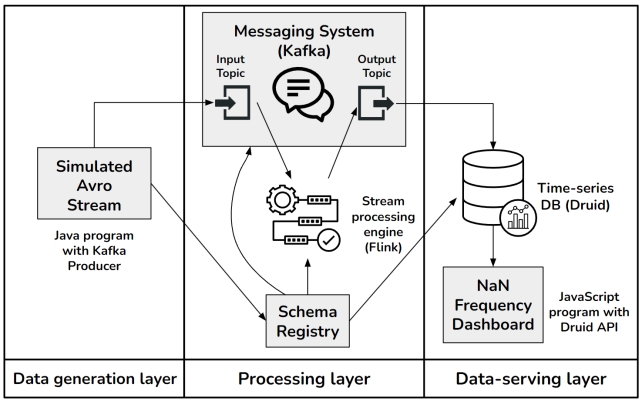


**Figure 1: Pipeline structure and its main components: Apache Kafka [2], Apache Flink [1], Apache Druid [7].**

## 2 Discussion of V-Mon properties

V-Mon's properties are assessed by setting up different configurations, with and without fault tolerance applied, to evaluate how requirement precedence affects performance. Exactly-once processing semantics are enforced to ensure data completeness.

Figure 2 presents example results for throughput and latency for each configuration, along with their corresponding saturation rates. Tests at varying parallelism levels show that the pipeline can adapt to different data rates, provided sufficient resources. Including fault tolerance (replicating data across Kafka brokers) noticeably impacts pipeline performance, reducing the saturation rate by 32.2 % and increasing average latency at this rate by 14.9 % when running at a parallelism level of 8. The fault-tolerant pipeline still manages to sustain data rates of 1–10 billion tuples/day, while achieving latencies of ~500 ms within the 99th percentile. These numbers exceed current demands of a use case at Volvo Trucks, where the need for more efficient data processing is growing rapidly.

Full details of the design analysis, implementation and discussions of the findings are available in the master's thesis report [5].

The code is available in the following repository: **https://www.github.com/CWTED/PoC-stream-processing-pipeline**.

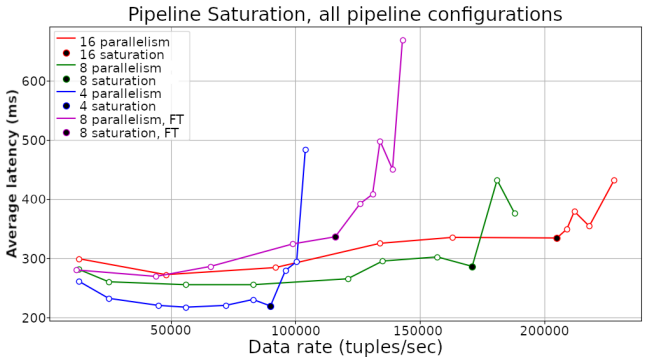


**Figure 2: Comparison of the scalability results from each of the four pipeline configurations.**

## 3 Conclusions

This work is about the design and construction of a stream processing pipeline, using a data validation and statistics process as an example of a stateful processing operator, with parallelism and fault tolerance features. The results, based on an example use case at Volvo Trucks, highlight the importance of careful capacity and requirement planning when designing data pipelines for Big Data settings. Moreover, the extensible design of the pipeline enables the support of feature extraction through, e.g., data synopsis and sketching, for further processing and analysis, as well as alternative data representations, e.g., knowledge graphs.

## Acknowledgments

The implementation of this work was done at Volvo Group Trucks Technology as part of the master's thesis of the first two authors. We acknowledge the support by the Swedish Research Council (Vetenskapsrådet) grant EPITOME (nr. 2021-05424).

## References

[1] Paris Carbone, Asterios Katsifodimos, Stephan Ewen, Volker Markl, Seif Haridi, and Kostas Tzoumas. 2015. Apache Flink™: Stream and Batch Processing in a Single Engine. *The Bulletin of the Technical Committee on Data Engineering* 38, 4 (2015).

[2] Ken Goodhope, Joel Koshy, Jay Kreps, Neha Narkhede, Richard Park, Jun Rao, and Victor Yang Ye. 2012. Building LinkedIn's Real-time Activity Data Pipeline. *IEEE Data Eng. Bull.* 35, 2 (2012), 33–45.

[3] Martin Hilgendorf, Vincenzo Gulisano, Marina Papatriantafilou, Jan Engström, and Binay Mishra. 2023. FORTE: an extensible framework for robustness and efficiency in data transfer pipelines. In *Proceedings of the 17th ACM Int"l Conference on Distributed and Event-based Systems*. 139–150.

[4] Haruna Isah, Tariq Abughofa, Sazia Mahfuz, Dharmitha Ajerla, Farhana Zulkernine, and Shahzad Khan. 2019. A Survey of Distributed Data Stream Processing Frameworks. *IEEE Access* 7 (2019), 154300–154316.

[5] Måns Josefsson and Carl-Magnus Wall. 2024. *Balancing Strict Performance Requirements and Trade-Offs for Efficient Data Handling in Unbounded Flows - Design Considerations for a Proof-of-Concept Stream Processing Pipeline for Vehicular Data Validation.* Master's thesis. Chalmers University of Technology and University of Gothenburg, Gothenburg, Sweden.

[6] Joris van Rooij, Johan Swetzén, Vincenzo Gulisano, Magnus Almgren, and Marina Papatriantafilou. 2018. eChIDNA: Continuous data validation in advanced metering infrastructures. In *2018 IEEE Int"l Energy Conference (ENERGYCON)*. 1–6.

[7] Fangjin Yang, Eric Tschetter, Xavier Léauté, Nelson Ray, Gian Merlino, and Deep Ganguli. 2014. Druid: A real-time analytical data store. In *Proceedings of the 2014 ACM SIGMOD int"l conference on Management of data*. 157–168.