INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING &TECHNOLOGY (IJCET)

ISSN Print: 0976-6367
ISSN Online: 0976-6375

Publishers of High Quality Peer Reviewed Refereed Scientific, Engineering & Technology, Medicine and Management International Journals

PUBLISHED BY

IAEME

IAEME Publication

Chennai, India

https://iaeme.com/Home/journal/IJCET

International Journal of Computer Engineering and Technology (IJCET)
Volume 17, Issue 2, March-April 2026, pp. 42-61, Article ID: IJCET_17_02_003
Available online at https://iaeme.com/Home/issue/IJCET?Volume=17&Issue=2
ISSN Print: 0976-6367; ISSN Online: 0976-6375; Journal ID: 5751-5249
Impact Factor (2026): 33.45 (Based on Google Scholar Citation)
DOI: https://doi.org/10.34218/IJCET_17_02_003

ARCHITECTING EVENT-DRIVEN DATA PIPELINES FOR REAL-TIME SUPPLY CHAIN DECISIONING

Hariprakash Pasumarthi
Senior Application Dev Analyst, BJS Wholesale Club, United States of America.

ABSTRACT

Modern supply chains operate in highly dynamic environments where disruptions, fluctuating demand, and global logistics complexities require organizations to make decisions in near real time. Traditional batch-oriented data processing architectures often introduce latency that limits the ability of enterprises to respond quickly to operational events such as shipment delays, inventory shortages, or supplier disruptions. Event-driven data pipeline architectures provide a scalable and responsive solution for enabling continuous data flow and real-time analytics across distributed systems.

This paper presents a generalized architectural framework for designing event-driven data pipelines that support real-time supply chain decisioning. The study examines the integration of streaming platforms, distributed messaging systems, microservices, and scalable data processing frameworks to build resilient and low-latency pipelines. It explores how event streams generated from logistics platforms, warehouse management systems, IoT sensors, and enterprise resource planning (ERP) systems can be processed and analyzed in real time to support operational intelligence.

The paper further discusses architectural considerations such as event schema management, fault tolerance, data consistency, scalability, and security in event-driven *ecosystems. A conceptual reference architecture is proposed to demonstrate how event ingestion, stream processing, and analytics layers interact to enable proactive decision-making. Additionally, the research highlights the role of modern data lakehouses, real-time dashboards, and machine learning models in transforming event streams into actionable supply chain insights.*

The findings suggest that event-driven data pipelines significantly improve supply chain responsiveness, reduce operational delays, and enhance predictive decision-making capabilities. By adopting event-driven architectures, organizations can achieve higher levels of supply chain visibility, resilience, and operational agility in increasingly complex global logistics networks.

Keywords: Event-Driven Architecture, Real-Time Data Pipelines, Supply Chain Analytics, Stream Processing, Distributed Messaging Systems, Real-Time Decisioning, Data Streaming Platforms, Operational Intelligence, Data Lakehouse, Microservices Architecture

Cite this Article: Hariprakash Pasumarthi. (2026). Architecting Event-Driven Data Pipelines for Real-Time Supply Chain Decisioning. *International Journal of Computer Engineering and Technology (IJCET)*, 17(2), 42-61.

DOI: https://doi.org/10.34218/IJCET_17_02_003

1. Introduction

Global supply chains have become increasingly complex due to globalization, digital commerce expansion, and growing customer expectations for faster delivery and real-time service transparency. Organizations today operate across geographically distributed suppliers, logistics providers, warehouses, and retail networks, generating massive volumes of operational data. This data originates from enterprise systems such as Enterprise Resource Planning (ERP), Warehouse Management Systems (WMS), Transportation Management Systems (TMS), Internet of Things (IoT) devices, and logistics tracking platforms. Effectively transforming this continuous stream of operational data into actionable insights is critical for maintaining supply chain resilience and operational efficiency.

Historically, supply chain analytics has relied heavily on batch-oriented data integration architectures. In these traditional architectures, data is periodically extracted, transformed, and loaded (ETL) into centralized data warehouses where analytical processing occurs. While such systems are suitable for historical analysis and reporting, they introduce significant latency between the occurrence of operational events and the availability of corresponding analytical insights. In rapidly changing supply chain environments, delays in data processing can result in missed opportunities for proactive decision-making, delayed responses to disruptions, and inefficient inventory or logistics management.

Recent advances in distributed computing, cloud infrastructure, and streaming technologies have enabled the development of event-driven data architectures. In event-driven systems, operational activities such as order creation, shipment updates, inventory movements, or sensor alerts are captured as events and transmitted through real-time data streams. These events can be processed immediately by scalable stream-processing frameworks, enabling organizations to detect anomalies, forecast disruptions, and trigger automated responses with minimal latency. As a result, event-driven architectures have emerged as a foundational approach for enabling real-time operational intelligence in modern supply chains.

Event-driven data pipelines leverage distributed messaging systems, scalable processing engines, and microservices-based architectures to continuously ingest, process, and analyze data streams. Unlike traditional batch pipelines, these architectures support asynchronous communication between systems, enabling high-throughput and fault-tolerant processing of real-time events. This capability is particularly valuable in supply chain environments where operational decisions such as rerouting shipments, adjusting production schedules, or reallocating inventory must be made quickly based on continuously evolving data.

However, designing effective event-driven data pipelines requires careful architectural considerations. Challenges such as event schema evolution, data consistency, fault tolerance, latency optimization, and system scalability must be addressed to ensure reliable and efficient data processing. Additionally, integrating real-time event streams with analytical platforms and decision-support systems requires robust data governance and orchestration strategies.

This paper presents a comprehensive exploration of architectural approaches for designing event-driven data pipelines that support real-time supply chain decisioning. The research examines key architectural components including event ingestion layers, distributed streaming platforms, real-time processing engines, and analytics integration frameworks. A conceptual architecture model is proposed to illustrate how these components interact to enable continuous data flow and intelligent operational decision-making.

By adopting event-driven data pipeline architectures, organizations can significantly enhance their ability to respond to supply chain disruptions, optimize logistics operations, and make data-driven decisions in real time. As supply chains continue to evolve toward digital and autonomous ecosystems, event-driven data architectures will play a central role in enabling intelligent, resilient, and adaptive supply chain operations.

2. Evolution of Supply Chain Data Architectures

The architecture of data systems supporting supply chain operations has evolved significantly over the past three decades. As supply chains expanded globally and digital technologies advanced, organizations progressively transitioned from static reporting infrastructures to dynamic, real-time data processing environments. Understanding this architectural evolution provides important context for the emergence of event-driven data pipelines as a modern solution for real-time supply chain decisioning.

2.1 Traditional Batch-Oriented Data Architectures

Early supply chain analytics platforms were primarily built around centralized data warehouses that relied on batch-based Extract, Transform, and Load (ETL) processes. In this architecture, operational data from systems such as Enterprise Resource Planning (ERP), Warehouse Management Systems (WMS), and Transportation Management Systems (TMS) was periodically extracted and consolidated into a centralized repository. Data transformation and aggregation processes prepared the data for reporting and business intelligence dashboards.

While batch-oriented architectures provided a structured approach to enterprise reporting, they introduced several limitations. Data updates were often performed at hourly, daily, or weekly intervals, creating latency between operational events and analytical visibility. As supply chain environments became increasingly dynamic, such delays reduced the ability of organizations to react quickly to disruptions such as shipment delays, supplier failures, or unexpected demand fluctuations.

Furthermore, traditional ETL pipelines were tightly coupled and difficult to scale. As the volume and variety of supply chain data increased especially with the rise of IoT sensors and digital logistics platforms batch systems struggled to handle the growing data velocity. These limitations created a demand for more responsive data architectures capable of continuous data processing.

2.2 Emergence of Distributed Data Platforms

The rise of distributed computing technologies in the early 2010s introduced new possibilities for large-scale data processing. Distributed storage systems and parallel processing frameworks enabled organizations to manage massive datasets across clusters of commodity hardware. These platforms supported scalable data lakes where structured, semi-structured, and unstructured supply chain data could be stored and analyzed.

Data lake architectures allowed enterprises to ingest diverse operational data sources, including shipment telemetry, sensor data from logistics equipment, supplier communications, and customer order streams. However, many early data lake implementations still relied heavily on batch ingestion and processing workflows. Although they improved storage scalability, they did not fully address the need for real-time operational analytics.

2.3 Transition to Streaming Data Architectures

To overcome the latency limitations of batch processing, organizations began adopting streaming data architectures. In these systems, operational data is captured and transmitted as continuous event streams rather than periodic batch files. Distributed messaging systems enable producers such as IoT devices, logistics platforms, or enterprise applications to publish events that can be consumed by downstream services and analytics platforms.

Streaming architectures introduced several key advantages for supply chain systems:

- **Low latency processing:** Low latency processing, enabling near real-time insights.
- **High throughput:** High throughput, allowing systems to process millions of events per second.
- **Scalable event distribution:** Scalable event distribution, supporting large numbers of producers and consumers.
- **Fault tolerance:** Fault tolerance, ensuring data reliability in distributed environments.

These capabilities allowed supply chain operators to monitor logistics activities continuously and detect operational anomalies as they occurred.

2.4 Emergence of Event-Driven Architectures

Building upon streaming technologies, event-driven architectures (EDA) further refined how systems interact with real-time data. In event-driven systems, changes in operational state such as order creation, inventory updates, shipment status changes, or sensor alerts are represented as discrete events. These events trigger downstream processing logic, enabling automated responses and intelligent decision workflows.

Event-driven architectures promote loose coupling between systems by enabling asynchronous communication through event streams. Instead of relying on tightly integrated point-to-point connections, systems communicate through event brokers that distribute events to multiple consumers. This design improves system scalability, resilience, and flexibility.

Within supply chain environments, event-driven architectures enable capabilities such as:

- Real-time shipment tracking and route optimization
- Automated inventory replenishment
- Predictive disruption detection

- Dynamic logistics coordination
- Instant operational alerts and notifications

2.5 The Role of Event-Driven Data Pipelines

Event-driven data pipelines integrate event streaming, distributed processing frameworks, and analytics platforms into a unified architecture capable of supporting real-time decisioning. These pipelines continuously ingest operational events, transform and enrich the data, and deliver insights to decision-support systems and automated workflows.

Compared with traditional data architectures, event-driven pipelines provide several advantages:

Table: Comparison of Batch-Based vs. Event-Driven Architecture

Feature	Batch-Based Architecture	Event-Driven Architecture
Data Processing Latency	High (hours or days)	Low (milliseconds to seconds)
System Coupling	Tightly integrated	Loosely coupled
Scalability	Limited	Highly scalable
Operational Visibility	Delayed	Real-time
Decision Responsiveness	Reactive	Proactive

As supply chain networks become more digitized and interconnected, event-driven data pipelines are increasingly recognized as a foundational component for enabling intelligent, data-driven supply chain operations.

3. Core Principles of Event-Driven Data Pipeline Architecture

Event-driven data pipelines form the backbone of modern real-time analytics systems in supply chain environments. These pipelines are designed to capture operational events as they occur, process them through distributed streaming systems, and deliver insights to decision-making platforms with minimal latency. Unlike traditional batch-oriented systems, event-driven architectures operate continuously, enabling organizations to respond immediately to supply chain disruptions, demand fluctuations, and logistics updates.

This section explores the fundamental architectural principles and components that define event-driven data pipelines.

3.1 Event Producers

Event producers are systems or devices that generate operational events within the supply chain ecosystem. These events represent state changes or activities occurring within business processes. Examples include order placement, shipment dispatch, warehouse inventory updates, supplier confirmations, and sensor readings from logistics equipment.

Event producers may include:

- Enterprise systems such as ERP and order management platforms
- Warehouse management systems and inventory tracking systems
- Transportation management platforms
- IoT sensors monitoring temperature, location, or equipment status
- External logistics or supplier data feeds

Each operational event is typically represented in structured formats such as JSON or Avro, containing metadata and contextual information required for downstream processing.

3.2 Event Brokers and Messaging Infrastructure

Event brokers serve as the central communication backbone in event-driven architectures. They enable asynchronous communication between producers and consumers by acting as an intermediary that receives, stores, and distributes event streams.

Key functions of event brokers include:

- Reliable event ingestion and buffering
- Topic-based event distribution
- Event persistence for replay and recovery
- Horizontal scalability to support high event throughput

Distributed messaging systems enable organizations to process millions of events per second while maintaining fault tolerance and system resilience. Event brokers also decouple producers and consumers, allowing multiple downstream services to independently consume the same event stream.

3.3 Stream Processing Engines

Stream processing engines analyze and transform event streams in real time. These engines perform operations such as filtering, aggregation, enrichment, anomaly detection, and pattern recognition on continuously flowing data.

Common real-time processing tasks in supply chain systems include:

- Monitoring shipment status updates
- Detecting delays or route deviations
- Calculating real-time inventory levels
- Identifying supply disruptions
- Aggregating operational metrics for dashboards

Stream processing frameworks enable distributed data computation across clusters, ensuring scalability and low-latency processing even under heavy workloads.

3.4 Data Storage and Persistence Layers

Event-driven architectures typically integrate multiple storage layers to support both real-time and historical data analysis. These layers include:

- **Streaming storage:** Streaming storage, used for temporary buffering and event replay
- **Operational databases:** Operational databases, used for real-time application queries
- **Data lakes or lakehouses:** Data lakes or lakehouses, used for large-scale historical analytics

This multi-layer storage approach ensures that both operational decision-making systems and analytical platforms can access relevant data with appropriate performance characteristics.

3.5 Event Consumers and Decision Systems

Event consumers are applications or services that subscribe to event streams and perform actions based on incoming data. Consumers may include automated workflows, alerting systems, predictive models, or real-time dashboards.

Examples of consumer-driven supply chain decision processes include:

- Automated shipment rerouting when delays are detected
- Inventory replenishment triggered by threshold events
- Supplier risk alerts triggered by delivery anomalies
- Dynamic demand forecasting updates

By enabling immediate responses to operational events, event-driven pipelines allow organizations to shift from reactive to proactive supply chain management.

Figure 1: Event-Driven Supply Chain Data Pipeline Architecture

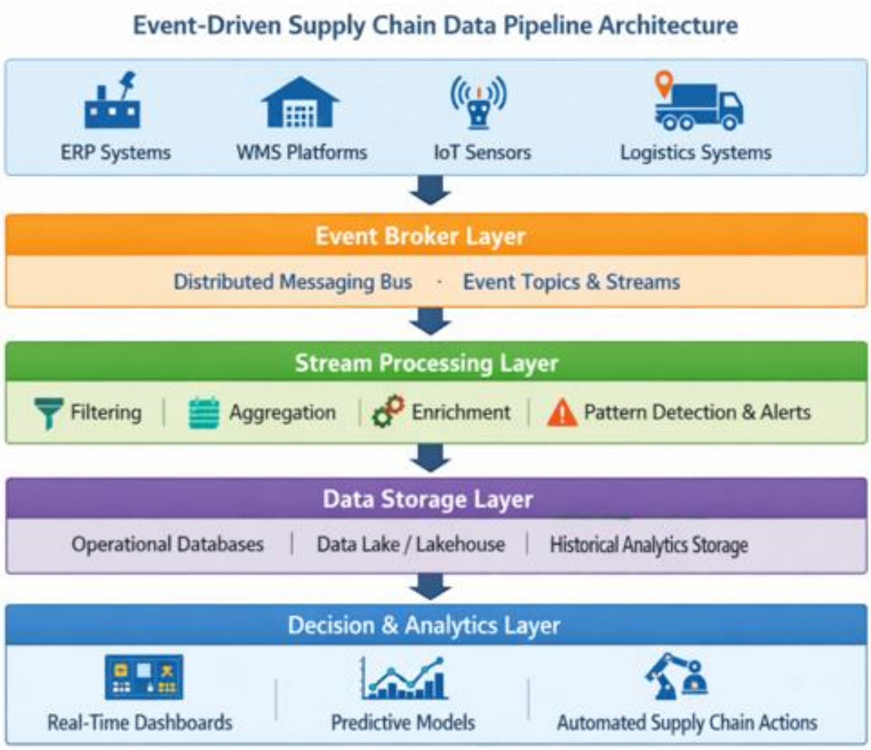

Fig. 1: Event-Driven Supply Chain Data Pipeline Architecture

Key Architectural Characteristics

Event-driven data pipelines typically exhibit several important characteristics:

- **Scalability:** Distributed architectures allow systems to handle rapidly growing event volumes.
- **Resilience:** Fault-tolerant messaging ensures events are not lost during system failures.
- **Low Latency:** Real-time processing enables rapid decision-making.
- **Loose Coupling:** Producers and consumers operate independently, improving system flexibility.
- **Extensibility:** New consumers or analytics services can easily subscribe to existing event streams.

These characteristics make event-driven architectures particularly suitable for supply chain systems that must process large volumes of operational events while maintaining high reliability and responsiveness.

4. Designing Scalable Event Ingestion and Streaming Layers

A critical component of event-driven data pipelines is the event ingestion and streaming layer, which is responsible for capturing, transporting, and distributing real-time operational events across distributed systems. In supply chain environments, this layer must support high event volumes, heterogeneous data sources, and strict reliability requirements. Designing a scalable ingestion and streaming infrastructure ensures that supply chain events are delivered efficiently to downstream analytics and decision systems.

4.1 Event Ingestion Mechanisms

Event ingestion refers to the process of collecting operational data from multiple supply chain systems and converting it into standardized event streams. These events originate from diverse sources such as transactional systems, logistics platforms, IoT sensors, and partner integrations.

Several mechanisms are commonly used to capture and ingest events:

- **API-based ingestion:** Applications publish events through REST or streaming APIs.
- **Change Data Capture (CDC):** Database changes are captured directly from transaction logs and converted into event streams.
- **IoT data streaming:** Sensor devices continuously transmit telemetry data such as temperature, location, and equipment status.
- **Application event publishing:** Microservices publish domain events when business operations occur.

Effective ingestion architectures must support high-throughput event intake while maintaining low latency and reliable delivery.

4.2 Distributed Messaging and Event Streaming Platforms

After events are ingested, they are transmitted through distributed messaging infrastructures that act as the backbone of event-driven systems. Messaging platforms organize events into logical channels known as topics or streams, allowing multiple applications to subscribe and process data independently.

Key design principles for event streaming platforms include:

- **Partitioned data streams:** Partitioned data streams for horizontal scalability
- **Persistent event logs:** Persistent event logs to support replay and recovery
- **Fault-tolerant distributed clusters:** Fault-tolerant distributed clusters to maintain system reliability
- **Event ordering guarantees:** Event ordering guarantees for consistent processing

In supply chain environments, messaging infrastructures enable thousands of concurrent producers and consumers to interact with real-time data streams without tightly coupling system dependencies.

4.3 Stream Partitioning and Scalability

To support high event volumes, streaming systems distribute data across multiple partitions. Each partition acts as an independent stream that can be processed in parallel by multiple processing nodes. Partitioning enables systems to scale horizontally as event traffic increases.

Effective partitioning strategies often rely on supply chain identifiers such as:

- Shipment ID
- Order ID
- Warehouse location
- Supplier identifier

By distributing events based on these keys, systems can maintain logical ordering while achieving parallel processing efficiency.

4.4 Schema Management and Data Governance

Because event-driven systems continuously exchange data between distributed services, maintaining consistent data structures is essential. Event schemas define the structure and format of each event message, including metadata and payload fields.

Schema management frameworks help ensure compatibility between producers and consumers by supporting:

- Version-controlled event schemas
- Backward and forward compatibility
- Validation of event formats

- Centralized schema registries

Effective schema governance prevents downstream processing failures and ensures data interoperability across the pipeline.

4.5 Reliability and Fault Tolerance

Supply chain systems often support mission-critical operations where data loss can lead to significant operational disruptions. Therefore, event ingestion and streaming architectures must incorporate strong reliability guarantees.

Key reliability mechanisms include:

- Event replication across cluster nodes
- Durable event storage to prevent message loss
- Acknowledgment-based delivery mechanisms
- Automatic failover and recovery processes

These mechanisms ensure that event streams remain available even in the presence of hardware failures or network interruptions.

Table 1: Comparison of Data Ingestion Approaches in Supply Chain Systems

Ingestion Method	Data Source Type	Latency	Typical Use Cases
Batch ETL	Databases, files	High	Historical reporting
API-Based Streaming	Enterprise applications	Low	Order processing events
Change Data Capture	Transaction databases	Low	Real-time database updates
IoT Streaming	Sensors and devices	Very Low	Equipment monitoring
Message Queue Ingestion	Distributed services	Low	Event-driven microservices

Key Benefits of Scalable Streaming Layers

A well-designed event ingestion and streaming layer provides several benefits for supply chain operations:

- **Real-time operational visibility:** Real-time operational visibility across logistics networks
- **Scalable data pipelines:** Scalable data pipelines capable of handling large event volumes
- **Improved system resilience:** Improved system resilience through distributed architectures
- **Enhanced integration:** Enhanced integration between enterprise systems and analytics platforms

These capabilities enable organizations to transform raw operational events into actionable intelligence that supports real-time supply chain decision-making.

5. Real-Time Stream Processing and Data Transformation

Once events are captured and transmitted through the streaming infrastructure, they must be processed and transformed into meaningful information that supports operational decision-making. Real-time stream processing enables continuous analysis of incoming data streams, allowing supply chain systems to detect patterns, generate alerts, and trigger automated actions. Unlike batch analytics, which processes large datasets periodically, stream processing evaluates events as they arrive, enabling near-instant insights.

In supply chain environments, real-time stream processing plays a crucial role in monitoring logistics operations, detecting anomalies, and optimizing operational workflows.

5.1 Stream Processing Workflows

Stream processing workflows consist of continuous computational tasks that operate on incoming event streams. These workflows analyze events sequentially or in parallel, performing various transformations and aggregations before delivering processed data to downstream systems.

Typical stream processing tasks include:

- Filtering irrelevant or duplicate events
- Aggregating metrics across operational streams
- Joining events from multiple data sources
- Enriching events with contextual information
- Detecting operational anomalies

For example, a logistics monitoring system may continuously analyze shipment tracking events to identify delays or route deviations in real time.

5.2 Event Enrichment and Data Integration

Raw supply chain events often lack sufficient contextual information to support decision-making. Event enrichment enhances event streams by integrating additional data from external systems or reference datasets.

Examples of enrichment processes include:

- Adding supplier information to purchase order events
- Integrating warehouse location data with shipment updates
- Combining sensor telemetry with logistics metadata
- Incorporating weather or traffic data into transportation analytics

Enrichment allows real-time processing systems to generate more comprehensive insights without requiring downstream systems to perform additional data lookups.

5.3 Window-Based Stream Analytics

Many supply chain analytics operations require aggregating data across specific time intervals. Window-based analytics enables stream processing engines to analyze events within defined time windows.

Common windowing techniques include:

- **Tumbling windows:** Tumbling windows, which process fixed non-overlapping intervals
- **Sliding windows:** Sliding windows, which continuously analyze overlapping event windows
- **Session windows:** Session windows, which group events based on periods of activity

For example, a warehouse monitoring system might compute the number of shipments processed within a five-minute window to detect abnormal operational patterns.

5.4 Real-Time Anomaly Detection

One of the most valuable capabilities of stream processing is real-time anomaly detection. By continuously analyzing event streams, processing systems can identify unusual patterns or deviations from expected operational behavior.

Examples of supply chain anomalies include:

- Unexpected shipment delays
- Sudden inventory shortages
- Equipment malfunctions detected by IoT sensors
- Supplier delivery disruptions

Advanced anomaly detection systems may integrate machine learning models that evaluate event patterns and generate predictive alerts.

5.5 Stream Processing Optimization

To ensure efficient real-time analytics, stream processing systems must be optimized for performance and scalability. Several design strategies are commonly used:

- **Parallel stream processing:** Parallel stream processing, distributing workloads across multiple nodes
- **Stateful processing:** Stateful processing, maintaining intermediate computation results
- **Event checkpointing:** Event checkpointing, preserving processing states for fault recovery
- **Load balancing:** Load balancing, distributing event streams across processing instances

These techniques allow organizations to maintain low-latency analytics even under high event volumes.

Figure 2: Real-Time Stream Processing Workflow in Event-Driven Supply Chain Systems

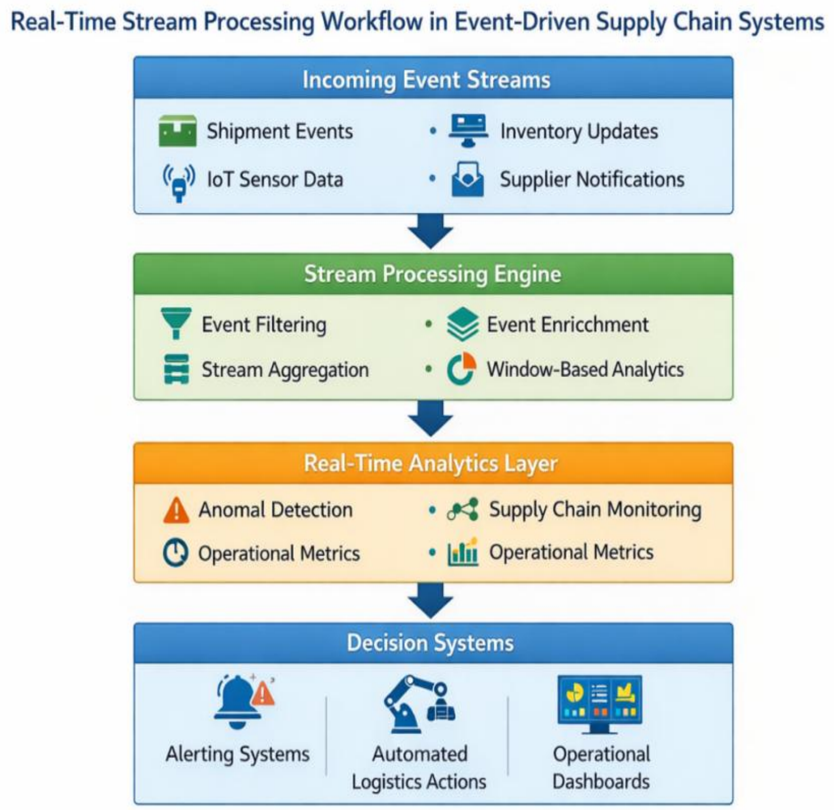

Fig. 2: Real-Time Stream Processing Workflow in Event-Driven Supply Chain Systems

Importance of Real-Time Processing in Supply Chains

Real-time stream processing enables supply chain systems to transition from passive monitoring to proactive operational management. By continuously analyzing event streams, organizations can detect disruptions earlier, reduce response times, and optimize resource allocation.

These capabilities significantly improve operational resilience, allowing enterprises to maintain supply chain continuity even under unpredictable conditions.

6. Data Storage, Lakehouse Integration, and Analytical Layers

In event-driven data architectures, the storage and analytics layers play a critical role in transforming real-time operational data into actionable intelligence. Supply chain systems generate large volumes of continuous event streams that must be stored, processed, and analyzed efficiently. Modern data architectures therefore combine real-time operational databases, scalable data lakehouses, and advanced analytical platforms to support both operational decision-making and long-term strategic analysis.

6.1 Operational Data Storage

Operational data storage systems support real-time applications that require immediate access to recently processed events. These systems typically store short-term transactional data used by logistics platforms, inventory management systems, and supply chain monitoring tools.

Key characteristics of operational storage systems include:

- Low-latency read and write performance
- High availability and fault tolerance
- Support for real-time application queries
- Integration with streaming data pipelines

Operational storage systems often maintain current supply chain states such as inventory levels, shipment status, and order processing metrics.

6.2 Data Lakes and Lakehouse Architectures

While operational systems manage real-time data access, long-term analytical workloads require scalable storage platforms capable of handling large historical datasets. Data lakes provide a flexible repository where structured and unstructured supply chain data can be stored in raw or processed formats.

However, traditional data lakes often lack governance and query optimization capabilities. To address these limitations, many organizations have adopted data lakehouse architectures, which combine the scalability of data lakes with the performance and structure of data warehouses.

Key advantages of lakehouse architectures include:

- Unified storage for structured and semi-structured data
- Scalable analytics for large datasets
- Support for machine learning and advanced analytics
- Improved data governance and metadata management

In supply chain environments, lakehouses enable organizations to store large volumes of historical logistics, inventory, and supplier data for predictive modeling and trend analysis.

6.3 Integration with Analytical Platforms

Analytical platforms transform stored data into insights through business intelligence dashboards, reporting tools, and advanced analytics systems. These platforms provide decision-makers with real-time visibility into operational performance across the supply chain.

Common analytical capabilities include:

- Real-time operational dashboards
- Supply chain performance monitoring

- Inventory and demand forecasting
- Supplier performance analysis
- Logistics optimization analytics

By integrating streaming pipelines with analytical systems, organizations can maintain a continuous flow of insights that support both tactical and strategic decisions.

6.4 Real-Time Data Visualization and Dashboards

Real-time dashboards provide operational teams with immediate visibility into supply chain activities. These dashboards display key performance indicators (KPIs) such as shipment delivery status, warehouse throughput, inventory availability, and transportation efficiency.

Key features of real-time supply chain dashboards include:

- Continuous data refresh from streaming pipelines
- Interactive visualizations for operational monitoring
- Alert generation for operational anomalies
- Integration with predictive analytics systems

These visualization tools enable managers to quickly identify disruptions and initiate corrective actions.

6.5 Predictive and Prescriptive Analytics

Beyond operational monitoring, advanced analytics systems apply predictive and prescriptive models to supply chain data. Predictive models use historical and real-time data to forecast potential disruptions, while prescriptive systems recommend optimal actions.

Examples include:

- Predicting shipment delays using historical logistics patterns
- Forecasting product demand using real-time order streams
- Optimizing warehouse operations through predictive inventory models
- Identifying supplier risks through anomaly detection models

These capabilities allow organizations to transition from reactive supply chain management to proactive and intelligent decision-making.

Table 2: Comparison of Storage and Analytics Layers in Event-Driven Data Architectures

Layer	Primary Purpose	Data Type	Typical Use in Supply Chain
Operational Databases	Real-time application access	Structured transactional data	Order processing, inventory updates
Streaming Storage	Temporary event buffering	Event streams	Real-time pipeline processing
Data Lake	Large-scale raw data storage	Structured & unstructured	Historical logistics data
Data Lakehouse	Unified analytics platform	Structured & semi-structured	Advanced analytics and ML
BI & Analytics Platforms	Data visualization and reporting	Aggregated analytics data	Supply chain performance monitoring

Importance of Integrated Data Architecture

A well-integrated storage and analytics architecture ensures that supply chain organizations can leverage both real-time operational insights and long-term analytical intelligence. Event-driven pipelines continuously feed data into storage systems, enabling organizations to maintain a comprehensive view of supply chain performance across operational, tactical, and strategic levels.

This integration significantly enhances supply chain visibility, forecasting accuracy, and operational resilience.

7. Security, Governance, and Reliability in Event-Driven Data Pipelines

As organizations increasingly rely on real-time data pipelines for supply chain decision-making, ensuring security, governance, and system reliability becomes a critical architectural requirement. Event-driven architectures process large volumes of sensitive operational data, including supplier transactions, logistics information, inventory records, and customer order details. Without robust security and governance mechanisms, such systems may be vulnerable to data breaches, operational disruptions, and compliance violations.

This section discusses key design principles for securing event-driven data pipelines and maintaining reliable operations in distributed environments.

7.1 Data Security in Streaming Architectures

Event-driven data pipelines rely on distributed messaging systems and multiple interconnected services. Securing these components requires a comprehensive security framework that protects data during transmission, processing, and storage.

Key security practices include:

- **End-to-end encryption:** End-to-end encryption to protect data in transit between event producers and consumers
- **Authentication mechanisms:** Authentication mechanisms to verify the identity of applications publishing or consuming events
- **Access control policies:** Access control policies to restrict access to event streams and data repositories
- **Secure API gateways:** Secure API gateways to protect event ingestion endpoints

These mechanisms help prevent unauthorized access and ensure the confidentiality of supply chain data.

7.2 Event Governance and Data Quality Management

Event governance ensures that event streams maintain consistent structure, quality, and traceability across the pipeline. Because multiple services interact through shared event streams, governance frameworks help maintain interoperability and prevent system failures caused by inconsistent data formats.

- □ Important governance practices include:
- □ Maintaining centralized event schema registries
- □ Implementing version control for event schemas
- □ Monitoring event quality and data integrity
- □ Enforcing data lifecycle policies

Effective governance ensures that supply chain data remains accurate, reliable, and compatible with downstream analytics systems.

7.3 Fault Tolerance and System Resilience

Event-driven pipelines must maintain continuous operation even when system components fail. Distributed streaming architectures therefore incorporate multiple fault-tolerance mechanisms to ensure system resilience.

Common resilience strategies include:

- □ Event replication across distributed nodes
- □ Automatic failover mechanisms for processing nodes
- □ Checkpointing and state recovery for stream processing engines
- □ Event replay capabilities using persistent event logs

These mechanisms allow organizations to recover quickly from failures without losing critical operational data.

7.4 Monitoring and Observability

Monitoring and observability are essential for maintaining the health and performance of real-time data pipelines. Continuous monitoring enables system operators to detect performance bottlenecks, processing delays, or abnormal event flows.

Observability frameworks typically track:

- □ Event throughput and processing latency
- □ Stream processing resource utilization
- □ Event delivery success rates
- □ System error rates and failure events

Operational dashboards and automated alerting systems help infrastructure teams quickly diagnose issues and maintain stable pipeline operations.

7.5 Compliance and Regulatory Considerations

Supply chain data pipelines often process information that must comply with regulatory and industry standards. Data governance policies must therefore ensure that event streams adhere to data protection regulations and organizational compliance requirements.

Compliance strategies may include:

- Data anonymization and masking techniques
- Secure audit trails for event processing
- Access logging and monitoring
- Data retention policies aligned with regulatory standards

By incorporating these controls, organizations can maintain regulatory compliance while operating large-scale real-time data systems.

CONCLUSION

The increasing complexity and digitalization of global supply chains have created a growing demand for real-time data processing capabilities. Traditional batch-oriented data architectures are no longer sufficient to support modern operational requirements, as they introduce latency that limits timely decision-making. Event-driven data pipelines provide a scalable and responsive alternative that enables organizations to process supply chain events continuously and transform them into actionable insights.

This paper explored the architectural principles behind event-driven data pipelines designed for real-time supply chain decisioning. It examined the evolution of supply chain data architectures, highlighting the transition from batch processing systems to distributed streaming platforms. The study discussed core architectural components including event producers, messaging infrastructures, stream processing engines, and multi-layer data storage architectures. These components collectively enable continuous event ingestion, real-time data transformation, and immediate analytics delivery.

The research also emphasized the importance of scalable streaming layers, real-time stream processing techniques, and integrated analytical platforms in enabling operational intelligence across supply chain networks. By incorporating advanced analytics and predictive models, organizations can identify potential disruptions earlier and respond proactively to operational challenges.

Additionally, the paper addressed critical architectural considerations related to system security, data governance, and pipeline reliability. Robust monitoring, fault tolerance mechanisms, and governance frameworks are essential to ensure that event-driven systems remain secure, resilient, and compliant with regulatory standards.

Overall, event-driven data pipeline architectures provide a powerful foundation for modern supply chain analytics. By enabling continuous data flow, low-latency processing, and intelligent automation, these architectures significantly enhance supply chain visibility, operational agility, and decision-making efficiency. As supply chains continue to evolve toward highly interconnected digital ecosystems, event-driven data platforms will play a central role in enabling resilient and intelligent supply chain operations.

REFERENCES

[1] H. Isah, T. Abughofa, S. Mahfuz, D. Ajerla, F. Zulkernine, and S. Khan, "A survey of distributed data stream processing frameworks," IEEE Access, vol. 7, pp. 154300-154316, 2019.

[2] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," Future Generation Computer Systems, vol. 97, pp. 219-235, 2019.

[3] S. Sankaranarayanan, M. Salehe, and R. Buyya, "Data flow and distributed deep neural network-based low-latency IoT-edge computation model for big data environments," Engineering Applications of Artificial Intelligence, vol. 94, 2020.

[4] E. Mehmood et al., "Challenges and solutions for processing real-time big data streams: A systematic literature review," IEEE Internet of Things Journal, vol. 7, no. 9, pp. 8132-8147, 2020.

[5] Y. Yang, J. Xu, and C. Zhang, "Big data-driven edge-cloud collaboration architecture for cloud manufacturing," IEEE Access, vol. 8, pp. 45938-45950, 2020.

[6] H. Hernandez, B. Xiao, and V. Tudor, "Enabling intelligent data pipelines for IoT-based application systems," in Proceedings of the IEEE International Conference on Pervasive Computing and Communications, 2020.

[7] S. P. Veluru, "Streaming data pipelines for AI at the edge: Architecting for real-time intelligence," International Journal of Artificial Intelligence, Data Science and Machine Learning, vol. 3, no. 2, pp. 56-71, 2021.

[8] S. Kumar Devarasetty, "Integrating AI and data engineering in IoT ecosystems: Streaming data management for smart devices," ComputerTech Journal, vol. 5, no. 2, pp. 61-72, 2021.

[9] S. Singu, "Designing scalable data engineering pipelines using cloud-based distributed processing frameworks," ESP Journal of Engineering and Technology Advancements, vol. 1, no. 2, pp. 176-187, 2021.

Citation: Hariprakash Pasumarthi. (2026). Architecting Event-Driven Data Pipelines for Real-Time Supply Chain Decisioning. International Journal of Computer Engineering and Technology (IJCET), 17(2), 42-61.

Abstract Link: https://iaeme.com/Home/article_id/IJCET_17_02_003

Article Link: https://iaeme.com/MasterAdmin/Journal_uploads/IJCET/VOLUME_17_ISSUE_2/IJCET_17_02_003.pdf

✉ **editor@iaeme.com**