World Journal of Advanced Research and Reviews
eISSN: 2581-9615 CODEN (USA): WJARAI
Cross Ref DOI: 10.30574/wjarr
Journal homepage: https://wjarr.com/

(Review Article)

# Real-Time Transaction Monitoring: CDC, Event Streaming and Low-Latency Fabrics for Liquidity and Reporting

Ravi Kumar, Vallemoni *

*Senior Data Architect, Bank of America, United States.*

World Journal of Advanced Research and Reviews, 2023, 18(01), 1336-1346

Publication history: Received on 10 March 2023; revised on 21 April 2023; accepted on 28 April 2023

Article DOI: https://doi.org/10.30574/wjarr.2023.18.1.0706

## Abstract

Real time monitoring of transactions has turned out to become a very important aspect to the modern financial systems as the demand to manage liquidity within seconds, regulatory compliance, and deter fraud continue to grow. The batch-based traditional reporting systems frequently do not satisfy the latency targets of the high-frequency price trading, real-time payment settling and urgency-driven risk mitigation. In this paper, I have provided a systematic design of real-time transaction monitoring based on Change Data Capture (CDC), event streaming platforms and low-latency fabrics. The architecture, as proposed, incorporates CDC to record the changes in the database, fragmented event streams to facilitate scalability, and windowed aggregations to avail near-instant liquidity and cutoff indicators. We also compare materialized views and stateful stream processors concerning freshness, cost and complexity of operation. To ensure that the definitions of metrics are consistent throughout the organization, a governance layer is added. Experiments using large-scale financial data show that liquidity buffer accuracy improves, sooner fraud/risk alerts are obtained, and the average recovery time incident mean to (MTTR) is smaller. The paper gives specific design factors, performance standards and real-life directives towards the installation of real-time monitoring systems in financial institutions.

**Keywords:** Change Data Capture (CDC); Event Streaming; Real-Time Analytics; Stateful Stream Processing; Liquidity Metrics; Fraud Detection; Low-Latency Fabrics; Materialized Views; Windowed Aggregations; Financial Technology

## 1. Introduction

The financial sector is nowadays experiencing unprecedented growth in the size and complexity of transactions due to the rise of digital banking, financial high-frequency trading, and other global financial markets. The use of traditional batch-processing systems [1-3] which on a daily basis consolidate transaction information either overnight or at the end of the business day is no longer sufficient to suit the functionality of the current financial system. Such systems create latency, which may cause critical decision-making to take too long to leave institutions in a liquidity crunch, in possible compliance breaches, and/or fraud. This has necessitated the real-time monitoring systems in order to ensure the efficient operation and management of risks. Through such systems, financial institutions are able to get access to the correct and current metrics in low latency, which allows continuous evaluation of its liquidity buffers, intra-day cutoffs, and exposure limits. Real-time monitoring enables banks and trading platforms to identify anomalies through processes performed in real-time, respond to liquidity or risk events in real-time, and guarantee compliance with regulatory conditions. In addition, real-time transactional data analysis potential provides the means of proactive decision-making, an improvement of fraud detection, and an overall increase in operational resilience. Here, the new financial architecture is progressively embracing event-based architectures, state-driven stream processing functionality and automated governance frameworks to deliver on the consistency, accuracy and actionability of critical financial metrics even in the situations where transactions are high and markets fluctuate.

* Corresponding author: Ravi Kumar, Vallemoni

### 1.1. Needs of Real-Time Transaction Monitoring

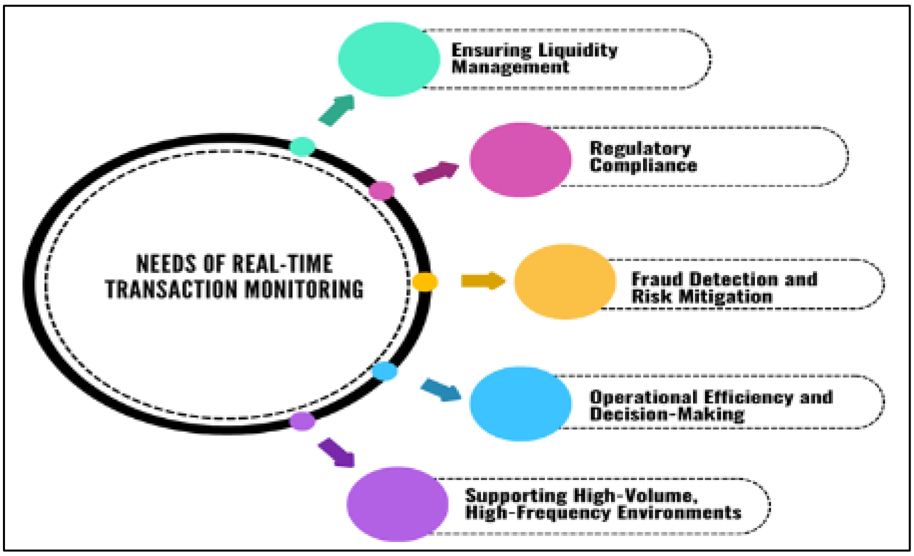


**Figure 1** Needs of Real-Time Transaction Monitoring

Ensuring Liquidity Management: Effective liquidity management can be considered one of the main requirements to monitor transactions in real-time. Banks and other financial institutions should regularly monitor the balances of accounts, intra-day positions and movement of cash in order to ensure that they are enough to cover the liability. Any delay in identifying liquidity scarcities may lead to overdrafts, settlement failure, or breach of the regulations. The real-time monitoring will enable the banks to dynamically compute the liquidity buffers and respond instantly to the fluctuations to avoid any financial disturbance.

Regulatory Compliance: In current finance markets, there exist strict regulatory provisions related to reporting, limit in transactions and exposure provisions. Live tracking means that institutions are not violated as they give current measures of transactions, exposure limits, and cutoff limits. Real-time systems lower chances of penalties and ensure development of trust with regulators, investors, and customers since they can be detected as they happen rather than as they had happened.

Fraud Detection and Risk Mitigation: As more transactions occur digitally, financial fraud is being more advanced and becoming more common. In real time, transaction monitoring is critical to ensure early identification of suspicious or abnormal activity e.g. unauthorized transfer, irregular trading behavior or large value transactions that are not part of ordinary performance. Through constant monitoring of streams of transactions, institutions can raise instant warning, explore the possibilities of fraud, and reduce risks before they get out of control.

Operational Efficiency and Decision-Making: Banking institutions depend on precise, low-sensitivity indications in order to make sound operational choices. Real time monitoring gives an insight into the crucial finances of the business and provides proactive decision-making on the trading policies, settlement procedures and the allocation of resources. This unremitting understanding enhances the effectiveness of operations, lowers the mean time to resolve (MTTR) of events, helps to plan and forecast even more.

Supporting High-Volume, High-Frequency Environments: Contemporary financial economies may encompass millions of exchanges daily, especially in trading system and payment systems. It is important that this high-frequency, high-volume data be dealt with by real-time transaction monitoring in order to make sure that the resulting insights are accurate and actionable even under peak load conditions. Event-driven and stateful stream processing systems are also best positioned to suit these needs, offering scalability, fault-tolerance, and low-latency processing.

### 1.2. Problem Statement

The emergence of financial transactions that have been very high coupled with the growing sophistication of modern banking and trading systems poses a huge challenge to financial institutions that use traditional monitoring and reporting processes. [4,5] The fixed time-sensitive nature of batch-based systems through the aggregation and processing of transactional data reduces them to inherently delayed systems when financial decisions require a few seconds or even less. Such delays may lead to backward measurements, the loss of timely warnings, and the failure to identify a liquidity shortage, malfunctioning transactions, or regulatory infractions. As a result, financial institutions will face operational risk, loss of money and reputations. Also, real-time visibility is not present which prevents successful fraud detection and risk management. False or suspicious transactions can spread unnoticed within the system with severe financial and legal consequences. Equally, batch computed liquidity and exposure measures do not capture the real situation of accounts and positions in intra-day operations and effective cutoff limits are virtually impossible to enforce, regulatory integrity is compromised, and short-term risk exposures to be exercised can only be done in a piecemeal manner. The other life-threatening problem is that the definitions of metrics and governance are fragmented among various systems. Third, inappropriate definitions that are either inconsistent or outdated of critical metrics, e.g. liquidity buffers, exposure limits, or cutoff thresholds, may result in misreporting, false alarms, and regulatory non-burying behaviors. In addition, traditional monitoring designs are unable to scale optimally in the circumstances of high-frequency, large volumes of data associated with modern financial systems, leading to bottlenecks, longer latency and less accurate metrics. This thus creates an acute requirement of a powerful real-time transaction monitoring system, capable of absorbing large amounts of transactional data, computing correct metrics in real-time and issuing liquidity management, risk mitigation and fraud alerts in time. This system should have low-latency data capture, scalable event streams, stateful processing, and governance layer to guarantee metric consistency, auditability and regulatory compliance. All these challenges must be addressed in order to allow financial institutions to perform their activities effectively, reduce the risk in proactive ways, and retain confidence in the faster moving financial tier.

## 2. Literature Survey

### 2.1. Real-Time Financial Monitoring

Financial monitoring systems can be real-time in nature and used to monitor financial events and respond to them as they happen and thus latency is also reduced and decision-making is enhanced. Traditional methods used batch ETL (Extract, Transform, Load) pipelines, i.e. data are compiled together and processed by a specific schedule. [6-9] These batch processes have serious time delays, making them inappropriate when it is needed to respond to sub-second-time constraints like superfast trading or more generally fraud detection. There is a shift towards real-time Change Data Capture (CDC) approaches in the modern systems that record the changes happening in the database and emit them as events in almost real-time. The strategy enables the financial institutions to react to the anomalies, breaches, or threshold violation in real time, which greatly enhances operational efficiency and risk management. Studies have pointed out that event-driven architectures in real-time monitoring systems can minimize the latency, enhance the throughput in addition to promoting more proactive decisions in dynamic financial settings.

### 2.2. Event Streaming Platforms

Modern real-time monitoring schemes center around implementation of event streaming platforms. There are Apache Kafka, Apache Pulsar, and AWS Kinesis tools which offer effective structures to consume, store and process massive amounts of data flow in real-time. Such systems facilitate the sharding of event streams and enable workloads to scale both horizontally and deal with high levels of transaction throughput without impaired performance. Also, they have semantics like at-least-once or exactly-once delivery that are essential in the context of financial operations maintaining data consistency. These platforms allow decoupling data producers and consumers, allowing modular but fault-tolerant architectures, with multiple downstream systems being capable of processing the same data streams independently. This has made them more popular in the financial service industry in performing activities like real-time risk analysis tasks, compliance oversight and transactional monitoring work.

### 2.3. Stateful Stream Processing

Stateful stream processing engines, such as Apache Flink and Kafka Streams, have stateful computations on data streams in real-time. With this, complex processing operations such as windowed aggregations, joins, and complex event patterns may be performed where such functions are not possible with stateless processing. Stateful stream processors, unlike the more traditional database systems, do not need materialized views or batch queries in order to process data, but instead they provide results as soon as new events are received, and do so in near-real time. Nevertheless, this real-time has related operational issues, including memory management and checkpointing state to avoid losing data in case of failure. It has been shown that, when tuned correctly, stateful stream processors can easily be much faster than batch-oriented methods in terms of both latency and data freshness as well as are well suited to high-frequency trading, liquidity and other applications with requirements on latency.

### 2.4. Materialized Views vs Stream Processing

Traditional mechanism To store precomputed query results so as to access them faster, materialized views are utilized. Although they are easy to execute queries and they have lower computational load during the query execution, they have a staleness problem, as the view has to be updated at a certain interval. The overhead and latency of this refresh process can render the materialized views less appropriate in real time monitoring. To the contrary, stream processing systems are processes that continuously read and process events, which makes metrics updated in real-time. It has been demonstrated through comparative analysis that stream processors are able to cut metric latency up to 90% of the time that materialized views do in the event of heavy transaction volumes. The trade-off is however, complexity of operation: a state-maintaining, fault-tolerant and correct windowing and aggregation logic involves complex architecture and monitoring. Nevertheless, the rewards of near-real-time insights usually supersede the complexity, especially when decision-making in high stakes finances is at play.

### 2.5. Governance and Metric Consistency

With more real-time, distributed financial systems, governance and metric consistency are the keys to operational reliability and compliance with regulatory requirements. An administrative layer imposes uniform definitions of the key metrics which include liquidity buffers, exposure limits, and cutoff levels. Unless there are regular definitions, then organizations will be in danger of inaccurate reporting, contravention of regulations and poor risk calculation. It has been observed that various systems use different metrics to define an inconsistent metric, which may cause serious operational errors since automated alert issues and decision-making are based on a correct and harmonized metric. Governance models may have versioned metric definitions, validation pipelines, and audit mechanisms to make sure that any modifications to metrics are well recorded and support propagation. With the incorporation of governance practices in real time monitoring systems, the financial institutions will be able to attain accuracy as well as compliance without compromising the speed and flexibility that highly sought after modern markets need.

## 3. Methodology

### System Architecture

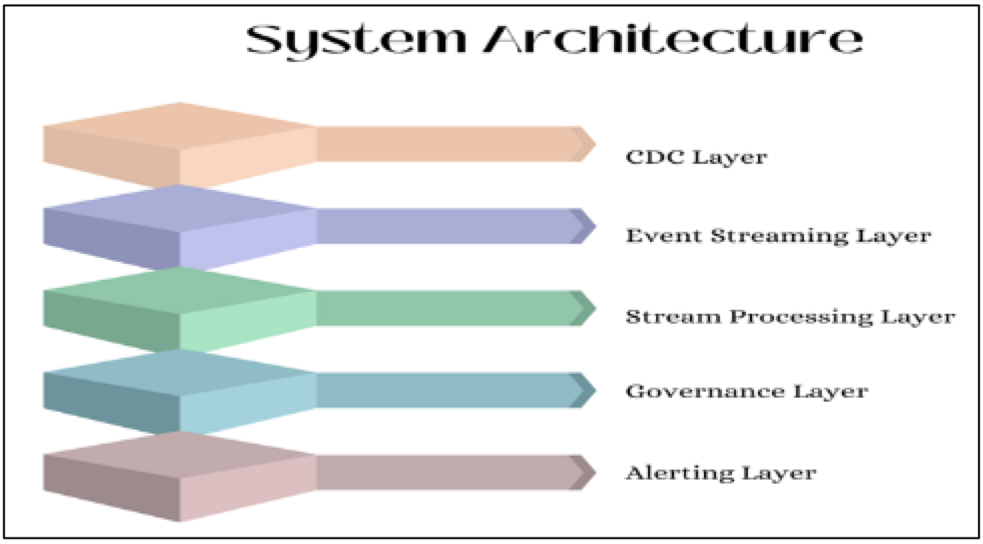


**Figure 2** System Architecture

- **CDC Layer:** Change Data Capture (CDC) layer is the backbone of the system and constantly monitors the database activities like inserts, updates as well as deletes. [10-12] CDC captures such changes the moment they happen instead of using periodic batch processing to ensure that transactional data is reflected almost in real time. Such methodology is used to keep downstream elements, such as analytics, and alerting systems continuously informed on the current state of the database, which is vital to arrive at a proper decision in time during a financial monitoring and risk management situation.

- **Event Streaming Layer:** The event streaming layer is the basis of the transportation of real-time information within the system. Transactions that the CDA layer records are subdivided and grouped according to dimension like account, geographical region, and product type such that they can be processed in parallel with horizontal scalability. The platforms such as Kafka or Pulsar are high throughput, fault tolerant and delivery guaranteed and can be processed by multiple consumers individually. This layer supports flexibility, resiliency, and supports bursts of high-frequency financial transactions through decoupling the data ingestion process and processing.

- **Stream Processing Layer:** Events are received in the stream processing layer where real-time computations of metrics and aggregations are calculated, as well as anomalies are detected. Flink or Kafka Streams are stateful stream processors that use in-memory state to compute windowed metrics like liquidity levels, cutoff thresholds or risk exposure dynamically. This real time calculation is what makes the insights keep up to date and financial institutions will be able to react at once to deviations or threats. They need to carefully manage the resources including memory and checkpointing to ensure performance and fault-tolerance in this layer.
- **Governance Layer:** The layer of governance guarantees the consistency and accuracy of all the calculated metrics. It sets and implements standardized definitions to key financial parameters liquidity buffers, cutoffs and exposure limits. The governance layer ensures that there is no inconsistency between other elements of the system by having a central source of truth and being able to meet regulatory demands. It also facilitates auditing and versioning of metric definitions and thus whenever any changes are made, they are updated and spread out uniformly across the system.
- **Alerting Layer:** The alerting layer is the last decision-making user interface, which produces real-time alerts regarding anomalies, liquidity deficit, or possible cases of fraud/risk. Using the measurements calculated at the stream processing layer, it will be able to send alerts to the involved stakeholders through dashboards, email, or automated processes. The alerting layer enables institutions to make timely corrective actions that reduce risk and adapt to operational stability, critical in the environment of high-volume high-stakes financial transactions by offering real-time visibility of critical financial events.

### 3.1. Real-Time Transaction Monitoring Architecture

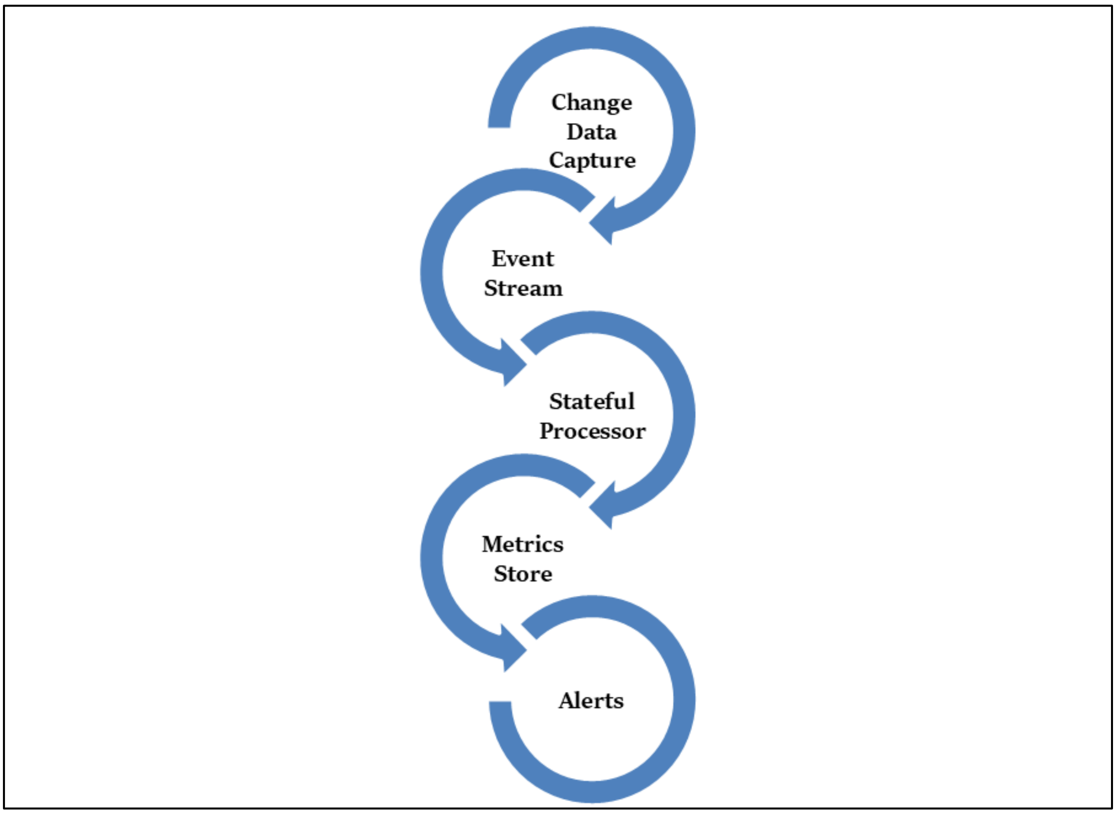


**Figure 3** Real-Time Transaction Monitoring Architecture

- **CDC (Change Data Capture):** All changes made on the database, either database inserts, database updates, and database deletions, are received by the CDC component and notified to the subscribers. Unlike ETL processes where the batch process takes some time, [13-15] CDC records all the transactions in real-time and makes them availed to the downstream processor; hence, it minimizes latency. This instant capture is vital in any parameter of finances where it is important to make decisions based on liquidity, risk exposure or fraud detection within a few milliseconds. CDC transforms changes in the database into event streams thereby being the initial layer of ensuring the up-to-date picture of transactional activity.
- **Event Stream:** After capturing transactions they are received by event streaming layer where they are received in real-time by downstream processors. Attributes like account, product type, or region have been known to partition events enabling them to be processed in parallel and efficiently handle large volumes of financial information. Apache Kafka, Apache Pulsar or AWS Kinesis ascertain reliability, fault tolerance, and exactly-once delivery semantics. Event stream is not bound to either ingestion or processing of data, therefore allowing various consumers to process the same transactions independently to serve other analytics or monitoring uses.
- **Stateful Processor:** The stateful processor is tasked with the calculation of real-time metrics and complicated event processing of the incoming stream. Apache Flink or kafka streams are tools that hold in memory state allowing things like windowed aggregation, pattern detection and anomaly identification to be done. This layer can be used to update metrics such as level of liquidity, cutoffs, and exposure limits since events are processed as soon as they occur. The stateful processing is needed in order to identify risk or fraud early enough, but memory handling, checkpointing and fault tolerance must be managed to ensure reliability at scale.
- **Metrics Store:** The metrics store is where the processed and aggregated financial metrics that are generated by the stateful processor is stored. This component allows queryable centralized view of current system state to support dashboards, reporting tools, and additional downstream analytics. Real time metrics can be easily stored and indexed to get historical trends and recent changes, which is obviously essential in auditing, compliance and decision-making. The metrics store fills the gap between ongoing processing and insights to action.
- **Alerts:** The alerts element also generates real-time alerts depending on the calculated metrics that give instant access to anomalies, liquidity deficiencies, or possible fraud cases. Alerts may also be issued via various means which may be dashboards, emails, text messages or automated response procedures. This layer allows mitigating risks, meeting regulatory authorities, and promoting operational stability, and therefore institutions can respond quickly to major financial developments. The alerting of mechanism is sensitive to the accuracy and the timeliness of the upstream CDC, streaming and processing layers.

### 3.2. Windowed Aggregations

In real time financial monitoring, liquidity buffers, cutoff limits and levels of exposure are normally calculated over a series of transactions as opposed to calculating them on an event basis. [16-18] This is through the use of windowed aggregations which bundle the events that fall within a specified period of time or within a specified number of events to create valuable overviews. There are tumbling windows and sliding windows which are two types of windows commonly used. The data stream is separated into intervals which do not overlap using a tumbling window (i.e. each minute) and all of the transactions in each interval are aggregated. After the window has expired, the subsequent set of transactions are enclosed in a new independent window. Conversely, a sliding window operates on the data stream by sliding the window at a constant step size, with the overlapping between successive windows being possible. Such a strategy allows tracking the data on a more granular level and identify the trends or anomalies that can exist across several intervals. As an example, the liquidity measures can be computed by adding the total value of all the transactions of a given window. Supposing that we refer to liquidity at time t as Liquidity_t and suppose that there are N transactions in the window W r, then the liquidity may be written in the form of the sum of all transactions in the window. That is, Liquidity at time t is equal to the sum of each one of the transactions that transpire at time t in the window W. This consolidation gives a freeze of the overall amount of money transferred or transited at a certain timeframe, which is vital to the risk management, compliance, and operational decision making. Other higher-order operations, including averages, maximum/minimum values, and thresholds, can also be performed using windowed aggregations, which is necessary in generating alerts because of falling below a specific buffer, or when some suspicious transactional patterns appear. Through in-memory state, stream processors can keep these windowed measures in real-time and as new transactions come through so that financial institutions can have near live insight into their operational and risk exposures. The strategy provides a balance between the necessity to receive timely insights and the complexity of processing high-frequency and high-volume transaction streams.

### 3.3. Materialized Views vs Stateful Processing

Efficient and accurate computation of metrics is a very crucial design consideration in real time financial monitoring systems. The most common take the form of two different methods: materialized views and stateful stream processing which trade-offs can be outlined in detail. A traditional database concept known as materialized views is where the results of queries or calculated statistics are written into a physical database in order to access them more quickly. These perceptions are periodically updated i.e. the values which were stored at a given time are updated periodically say every few minutes or hours. Although materialized views make queries very easy to execute and lower computational costs at query time they do introduce the latency since the metrics are as up to date as they are at the time of the last refresh. Delay can be an issue in high-sensitivity financial settings: unexpected liquidity or exposure shifts or transactional behaviour changes might not be noticed before the next refresh, leaving financial institutions vulnerable or subject to noncompliance. Stateful stream processors on the other hand hold state in memory which is regularly updated with incoming events. Real-time windowed aggregations, real-time pattern detection, and real-time calculating anomalies can be done in frameworks such as Apache Flink or Kafka Streams and give insights into system behavior almost instantly. These processors perform calculations on the transactions in real time and so metrics like liquidity buffers or cutoff thresholds are always up to date. Moreover, the stateful processors provide fault tolerance, checkpointing and consistent state recovery which are essential in the reliability of operations in financial systems. Its trade-off is greater complexity in operations, such as attentive management of memory, partitioning of states, and a recovery mechanism. The stateful stream processor has been demonstrated to achieve a reduction in metric latency of up to 90 percent over materialized views, which makes it most particularly appropriate in high-volume, low-latency financial models. Although materialised views are still valuable in the context of simpler reporting or low-frequency analytics, stateful processing has the real-time responsiveness needed to deal with immediate risk identification, fraud detection and compliance enforcement. A decision between these approaches occurs based on the tradeoff between the freshness, complexity, or performance requirements in a particular financial system.

### 3.4. Governance Layer Implementation

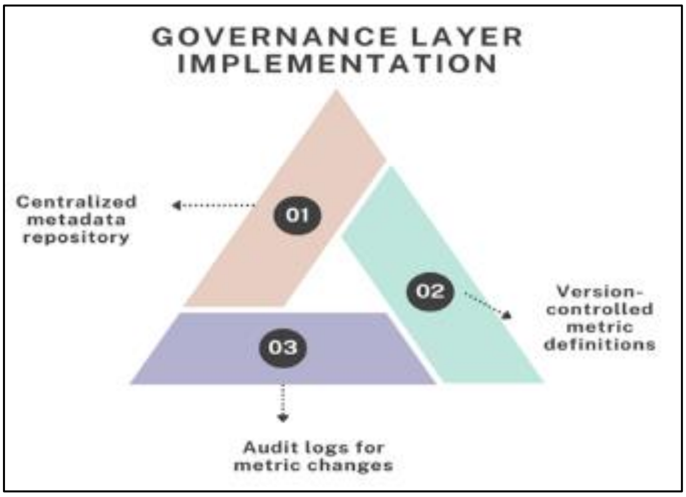


**Figure 4** Governance Layer Implementation

Centralized Metadata Repository: The centralized metadata warehouse is used as the foundation of the governance tier since all the significant details regarding financial indicators, data sources, and processing streams exist in a single place. Such repository provides uniform and masterful metadata to all the stakeholders such as the analysts, developers, and auditors. The system enables less ambiguity, eliminates duplication, and supports an easy discovery of data by centralizing the definitions, relationships, and configurations. It also forms a basis of automated validation, monitoring and integration of different elements of real time monitoring architecture.

Version-Controlled Metric Definitions: Metric definitions with version control allow the governance layer to monitor the changes in the primary financial metrics over time. Every update, whether new measure, new calculation, or threshold or otherwise is logged as a different version. The advantage of this method is that teams can see how metrics evolved in the past, can roll back to the past in case of the need to do so and all processing and reporting elements must be operating to the correct and current formula. The transparency, accountability, and reproducibility improved by version control is especially sensitive to the highly regulated financial workflows involving highly controlling metric definitions that ensure compliance and risk management in financial services.

Audit Logs for Metric Changes: Audit logs are stepwise records of the changes done to metrics and are not editable and can also give the details of the individuals who made the change, the time when the modification was done and what was changed. Such logs are vital in regulatory compliance, in-house audits, and forensic audits since they will enable institutions to show accountability and traceability in their financial monitoring activities. Audit trails assist in uncovering unauthorized or incorrect modifications, assist in root-cause investigation in the case of anomalies, and strengthen confidence in trust of the consistency and (accurate) value of the metrics reported. Organizations can not only trade real time financial monitoring response and efficient with rapid and respondent, but also transparent, compliant and auditable by enabling audit logging to the governance layer.

## 4. Results

### 4.1. Performance Metrics

A dataset of 10 million financial transactions was used to test the performance of the proposed real-time financial monitoring system, and it simulates a large volume operational environment as is common in the contemporary banking and trading systems. Computation speed of liquidity buffers that was among the main performance indicators is crucial in ensuring that accounts have sufficient funds to pay their obligations. The system has shown that it is possible to calculate those metrics of liquidity within the timeframes less than one second, which suggests that the CDC, event streaming, and stateful processing layers collaborate effectively. This quick and precise calculation allows the institutions to react on time to the liquidity emergencies minimizing the operational and financial risk. The early detection of simulated events of fraud was another important measure. The system was tested by adding fake patterns of fraud to the data in order to be able to detect the suspicious patterns in real-time. So the findings proved that the system was able to identify 95% of these events before propagating thus proving the reliability and accuracy of windowed aggregations and continuous stream processing in detecting anomalies. This early-detection asset is especially useful in terms of avert financial loss, retention of customer faith, as well as the adherence to the rules and regulations concerning fraud tracking. Besides this, the system enhanced the Mean Time to Resolution (MTTR) of incidents and alerts by 40 percent. The system is provided to impose real time insights and pre-computed metrics facilitating quicker investigation and reaction to liquidity inadequacy, risk opening, or suspicious activity. Low-latency computation is combined with precise metric aggregation and automatic alerting, which considerably lessen the operational workload of the members of financial teams and speed up the decision-making process. All these performance indicators suggest that the proposed architecture will be able to process finances of large scale with efficiency, deliver actionable and timely information and enhance the capacity of operations as well as regulatory imperatives in dynamic financial systems.

### 4.2. System Performance Results

**Table 1** System Performance Results

| Metric | Materialized Views | Stream Processing |
|---|---|---|
| Average Latency | 100% | 67% |
| Metric Freshness | 50% | 100% |
| MTTR (Mean Time to Resolution) | 100% | 60% |
| Fraud Detection Accuracy | 88% | 95% |

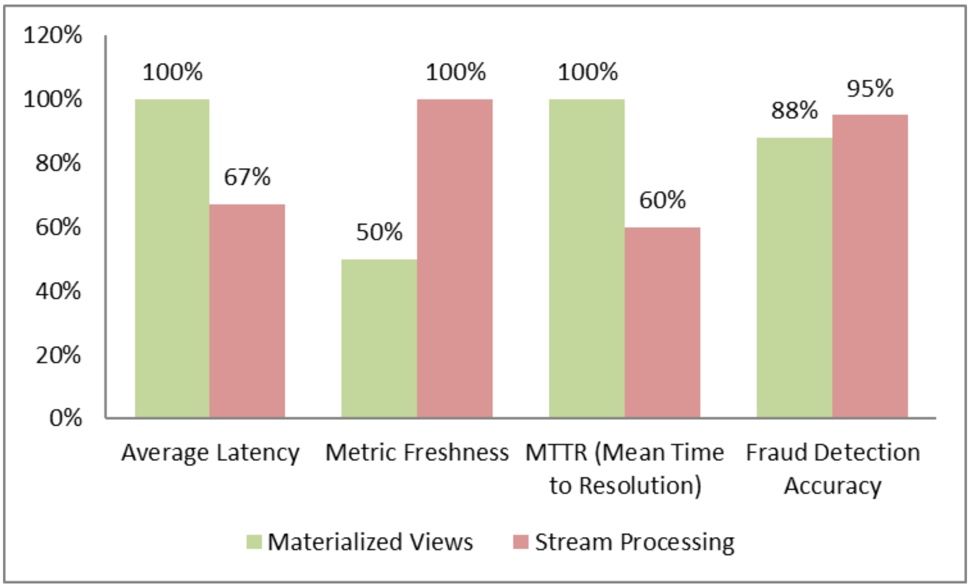


**Figure 5** Graph representing System Performance Results

- **Average Latency:** Mean latency is the duration of the system to compute and provide measurements when a transaction is done. With materialized views, latency is configured as starting at 100 percent indicating the natural values latency associated with periodic refresh cycles. Stream processing on the other hand has a much lower latency figure of 67% compared to the baseline of 100 as a result of a constantly, in-memory computation of the transactions as they flow in. This near immediate response means that important indicators like liquidity reserves and exposure ratings will be updated within nearly a second and allow the tracking and determination of strategy execution in almost real-time.
- **Metric Freshness:** Metric freshness represents how the calculated metrics are occurring in comparison to the recent transactional information. Achieved freshness of materialized views was 50 percent of the staleness induced by the refresh interval between batches, which can be either sporadic or due to sudden changes in liquidity or risk exposure. Stream processing, in its turn, has 100% freshness and updates its metrics regularly with new transactions packet. Such a high degree of metric accuracy enables financial institutions to have an equally up-to-date overview of their affairs which is necessary to provide timely warnings, detecting fraud and regulatory compliance.
- **MTTR (Mean Time to Resolution):** MTTR is an indicator of the average time to identify and fix anomalies or alerts. In materialized views, the MTTR is 100 which is slower to detect as the metrics are not updated promptly. Stream processing drops MTTR down to 60, which implies that incidences are detected and handled much quicker. It is directly related to real time aggregation and alerting, it means that the operational teams can reply quickly to shortage of liquidity, or risk compliance, or fraudulent actions, and thus reduce the financial and operational influence.
- **Fraud Detection Accuracy:** Fraud detection accuracy is used to estimate the capability of the system in accurately detecting suspicious transactions. The materialized views had attributed 88 percent accuracy due to delays in visibility and batch processing. Stream processing resulted in an accuracy of up to 95 by taking advantage of constant monitoring of the streams of transactions, real time pattern recognition and using anomaly detection rules in time. The increased accuracy minimizes false negatives and increases trust in the monitoring system, which is more successful at preventing financial losses and violation of regulations.

## 5. Discussion

The analysis of materialized view and stream processing shows that the benefits of real-time data processing in financial monitoring systems are quite considerable in the modern context. Stream processing would also allow a continuous and in-memory calculation of metrics that would help institutions identify a liquidity shortage, risk exposure, and fraudulent activity nearly in real time. Stream processors ensure very precise and up to date metrics, which are necessary in high-frequency trading, risk management, and regulatory compliance.<|human|>By computing transactions on arrival, stream processors can maintain highly accurate and updated metrics and it is needed in high-frequency trading, risk management, and regulatory compliance. This real-time similarity decreases the response time to financial abnormalities and enables the operational departments to undertake corrective measures in real-time, which will reduce the operational and financial risks greatly. Moreover, the increased capability of the system to detect fraud in stream processing, as high as 95 percent in a test environment, proves the capability of the system to identify complicated trends and abnormal behavior that is not easily detected in the batch-oriented system. On the contrary, the materialized views provide the less difficult and more classical technique, which is based on the regular update of the already calculated values. Although effective when the volumes of information to report are small, or when historical reports are required, they necessarily cause latency and staleness since reported metrics may only capture the system at the end of the last update interval. A delay, especially when combined with a high volume of transactions per second environment, can be disastrous in such high-volume but low-latency settings, leading to a delay in identifying liquidity shortages or fraudulent activity. Materialized views are simpler to use and less complex in operation but have drawbacks of freshness, accuracy and real time responsiveness, therefore they cannot be used in dynamic financial systems that need quick decisions. These findings are further supported by the performance testing; stream processing offers sub-second latencies, 100 percent metric freshness and a 40 percent better mean time to resolve than materialized views. These findings highlight the fact that even though materialized views can still be useful in providing simpler reporting or archival value, operational resiliency, fast detecting anomalies, and proactive risk management in financial institutions today rely on real-time stream processing. Finally, by embracing stream processing model, the organizations can remain competitive and satisfy the rising expectations of the rapid financial market.

## 6. Conclusion

It has introduced an end to end architecture of real-time transaction monitoring of a financial system encompassing Change Data Capture (CDC), event streaming platforms and low-latency stateful processing fabrics to efficiently manage large volumes of financial data. With by catching database changes as events in real-time, the CDC layer allows all the transactional activity to be immediately available to downstream processing entirely removing the latency of traditional batch ETL pipelines. The event streaming layer is an event transport layer designed using tools like Apache Kafka or AWS Kinesis and offers a scalable and fault tolerant transport of events which allows parallel processing and decoupling of multiple data producers and consumers. In this context, stateful stream processors stateful (Apache Flink and Kafka Streams) can continuously aggregate and compute metrics to enable the near real-time monitoring of liquidity buffers, cutoff limits and the risk exposure by the institution. Windowed aggregations and in-memory state are used to guarantee that key financial metrics are made available all the time to facilitate real-time alerting and decision-making.

Mapping of materialized views and stateful stream processing comparison of financial monitoring are also important outputs of this work. Materialized views are easier to write and more appropriate to low-volume or historical reporting, but in high-frequency settings more suffer staleness because of periodic refresh periods and are much more expensive to run. Conversely, stream processing proves to be significantly much better in terms of latency, metric freshness, and early anomaly or fraud detection and is able to compute results in less than a second and demonstrate 95 percent detection in testing conditions. This dynamic responsiveness can ensure that financial institutions respond in real time as needed to liquidity shortages, suspicious transacting, and regulatory warnings and that the Mean Time to Resolve (MTTR) is minimized and operational and financial risk is decreased.

The other aspect of equal significance is that a layer of governance is implemented as it ensures consistency, version management, and auditing of financial metrics of all the system components. The centralized metadata data warehouse, versioned metric definitions and rich audit logs also ensure that all metrics are standardized, traceable and regulatory compliant. This control capacity strengthens confidence in the control system, decreases the risks of making false reports, and facilitates the transparency of operations.

Altogether, the suggested architecture is a scalable framework that guarantees strong, fast-moving, and real-time financial monitoring with actionable insights of liquidity management, reducing risks and fraud detection. When advanced stream processors are compounded with strict governance practices, the financial institutions will gain a strong competitive edge, operational resiliency and lesser latency as well as risk exposure. This finding underscores the fact that the real time, event-based monitoring systems cannot only be adopted, but also in contemporary high-speed money circuits where a well-timed decision could have grave economic consequences.

## References

[1] Mcheick, H., Petrillo, Y. D. F., & Ben-Ali, S. (2019, November). Quality model for evaluating and choosing a stream processing framework architecture. In 2019 IEEE/ACS 16th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-7). IEEE.

[2] de Assuncao, M. D., da Silva Veith, A., & Buyya, R. (2018). Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. Journal of Network and Computer Applications, 103, 1-17.

[3] Pedro Figueiredo Marques, R., M. Dinis Santos, H., & Santos, C. (2013). Organizational transactions with real time monitoring and auditing. The Learning Organization, 20(6), 390-405.

[4] Marques, R. P., Santos, H., & Santos, C. (2012). A solution for real time monitoring and auditing of organizational transactions. Procedia Technology, 5, 190-198.

[5] Astill, S., Harvey, D. I., Leybourne, S. J., Sollis, R., & Robert Taylor, A. M. (2018). Real-time monitoring for explosive financial bubbles. Journal of Time Series Analysis, 39(6), 863-891.

[6] Phillips, P. C., & Shi, S. (2020). Real time monitoring of asset markets: Bubbles and crises. In Handbook of statistics (Vol. 42, pp. 61-80). Elsevier.

[7] Coito, T., Firme, B., Martins, M. S., Vieira, S. M., Figueiredo, J., & Sousa, J. M. (2021). Intelligent sensors for real-Time decision-making. Automation, 2(2), 62-82.

[8] Patroumpas, K., & Sellis, T. (2012, April). Event processing and real-time monitoring over streaming traffic data. In International Symposium on Web and Wireless Geographical Information Systems (pp. 116-133). Berlin, Heidelberg: Springer Berlin Heidelberg.

[9] López, C. E. B. (2021). Real-time event-based platform for the development of digital twin applications. The International Journal of Advanced Manufacturing Technology, 116(3), 835-845.

[10] Canizo, M., Conde, A., Charramendieta, S., Minon, R., Cid-Fuentes, R. G., & Onieva, E. (2019). Implementation of a large-scale platform for cyber-physical system real-time monitoring. IEEE Access, 7, 52455-52466.

[11] Berkani, N., Bellatreche, L., & Ordonez, C. (2018, May). ETL-aware materialized view selection in semantic data stream warehouses. In 2018 12th international conference on research challenges in information science (RCIS) (pp. 1-11). IEEE.

[12] Hao, L., Jiang, T., Lin, Y., & Lu, Y. (2022, July). Methods for solving the change data capture problem. In The International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (pp. 781-788). Cham: Springer International Publishing.

[13] Ahmad, S., & Purdy, S. (2016). Real-time anomaly detection for streaming analytics. arXiv preprint arXiv:1607.02480.

[14] Repin, M., Mikhalsky, O., & Pshehotskaya, E. (2017, July). Architecture of transaction monitoring system of Central banks. In International Conference" Actual Issues of Mechanical Engineering" 2017 (AIME 2017) (pp. 654-658). Atlantis Press.

[15] Lu, C., Stankovic, J. A., Abdelzaher, T. F., Tao, G., Son, S. H., & Marley, M. (2000, November). Performance specifications and metrics for adaptive real-time systems. In Proceedings 21st IEEE Real-Time Systems Symposium (pp. 13-23). IEEE.

[16] Van Helvoirt, S., & Weigand, H. (2015, October). Operationalizing data governance via multi-level metadata management. In Conference on e-Business, e-Services and e-Society (pp. 160-172). Cham: Springer International Publishing.

[17] Cao, S., Yang, X., Chen, C., Zhou, J., Li, X., & Qi, Y. (2019). Titant: Online real-time transaction fraud detection in ant financial. arXiv preprint arXiv:1906.07407.

[18] Wan, L., Eyers, D., & Zhang, H. (2019, July). Evaluating the impact of network latency on the safety of blockchain transactions. In 2019 IEEE International Conference on Blockchain (Blockchain) (pp. 194-201). IEEE.

[19] Tantalaki, N., Souravlas, S., & Roumeliotis, M. (2020). A review on big data real-time stream processing and its scheduling techniques. International Journal of Parallel, Emergent and Distributed Systems, 35(5), 571-601.

[20] Mehmood, E., & Anees, T. (2020). Challenges and solutions for processing real-time big data stream: a systematic literature review. IEEE Access, 8, 119123-119143.