International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

Architecting Near Real Time Data Integration Pipelines with PowerExchange and IICS Streaming

Srujana Parepalli

Senior Data Engineer, USA

ABSTRACT: By February 2019, enterprises operating large scale transactional platforms increasingly required data integration architectures capable of delivering near real time data movement with strong reliability and minimal impact on source systems. Traditional batch oriented extract transform load processes, while still widely used, were insufficient for use cases involving operational analytics, customer facing applications, regulatory reporting, and downstream system synchronization. Organizations sought integration solutions that could propagate data changes continuously while maintaining transactional integrity and operational stability. Change Data Capture had become a well established technique for addressing these requirements by enabling incremental data movement based on committed changes in source systems. Informatica PowerExchange was widely adopted in enterprise environments as a log based CDC solution capable of capturing database changes with low latency and minimal overhead. At the same time, Informatica Intelligent Cloud Services introduced streaming ingestion and processing capabilities that allowed CDC events to be transported and consumed in near real time within cloud based integration pipelines. This paper examines near real time data integration architectures that combine PowerExchange for change capture with IICS Streaming for event transport and processing, as understood and implemented by early 2019. The discussion focuses on architectural patterns for integrating on premises transactional systems with cloud based targets, emphasizing reliability, ordering, scalability, and operational governance. Particular attention is given to how CDC streams were modeled, buffered, and consumed to support continuous data propagation without tightly coupling source and target systems. Finally, the paper synthesizes design considerations and operational trade offs associated with PowerExchange and IICS Streaming based integration pipelines. These considerations include latency management, failure recovery, schema evolution handling, and monitoring of end to end data flow health. The intent is to provide a practical and historically grounded view of near real time data integration patterns that were actively adopted by enterprises in February 2019.

KEYWORDS: Near real time data integration, change data capture, Informatica PowerExchange, IICS Streaming, log based replication, enterprise data integration, cloud data pipelines, asynchronous data movement, transactional consistency, streaming ingestion. These keywords capture the technical and architectural focus of enterprise data integration practices as of February 2019, emphasizing continuous data movement, hybrid on premises and cloud architectures, and reliable propagation of transactional changes across distributed systems.

I. INTRODUCTION

By February 2019, enterprise data integration had entered a phase where near real time data availability was no longer a specialized requirement limited to niche use cases, but a mainstream expectation across multiple business domains. Organizations operating large transactional systems increasingly depended on timely data propagation to support operational dashboards, customer engagement platforms, downstream microservices, and regulatory reporting pipelines. Delays inherent in traditional batch processing models constrained responsiveness and introduced misalignment between operational systems and analytical or consuming platforms. Historically, batch oriented extract transform load processes were designed to optimize for throughput and stability rather than immediacy. These processes are typically executed on fixed schedules, extracting large volumes of data during predefined windows and loading results into downstream systems after substantial delay. While effective for periodic reporting and historical analysis, this approach proved inadequate for scenarios requiring up to date data representation, particularly in environments where transactional activity was continuous and high volume. Attempts to increase batch frequency often led to increased operational complexity and greater contention on source systems.

The growing adoption of cloud based platforms further intensified the need for continuous integration patterns. Enterprises increasingly operated in hybrid architectures where core transactional databases remained on premises while analytics, integration, and downstream applications migrated to cloud environments. This architectural split

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

required integration mechanisms that could bridge environments reliably, securely, and with predictable latency. Near real time integration emerged as a means of maintaining coherence across distributed systems without imposing intrusive access patterns on production databases. Change Data Capture provided the technical foundation for addressing these integration challenges by enabling incremental data movement based on committed transactions. Rather than querying source tables repeatedly, CDC mechanisms observed changes directly from database logs, preserving transactional ordering and minimizing overhead. Informatica PowerExchange became a common choice in enterprise environments due to its mature support for log based CDC across multiple database platforms and its integration with existing Informatica data integration tooling.

Dimension	Traditional Batch ETL	Near Real Time CDC Integration
Data movement	Periodic bulk loads	Continuous incremental changes
Latency	Hours to days	Minutes to near continuous
Source system impact	High during extract windows	Minimal via log based capture
Failure recovery	Re run full batch	Resume from log position
Scalability	Limited by batch window	Scales with throughput
Operational model	Scheduled jobs	Always on pipelines

At the same time, Informatica Intelligent Cloud Services expanded the integration landscape by offering streaming ingestion and processing capabilities in a managed cloud environment. IICS Streaming enabled CDC events captured by PowerExchange to be transported and processed continuously, supporting low latency delivery to cloud targets and downstream systems. This combination allowed enterprises to construct near real time integration pipelines that aligned with hybrid deployment models and operational governance requirements. This paper explores near real time data integration architectures built using PowerExchange and IICS Streaming as they were practiced by February 2019. It focuses on the architectural motivations, design patterns, and operational considerations that shaped these implementations, providing a grounded view of how enterprises addressed latency, reliability, and scalability in modern data integration workflows.

II. DRIVERS FOR NEAR REAL TIME DATA INTEGRATION IN ENTERPRISE ENVIRONMENTS

By February 2019, several converging factors drove enterprises to adopt near real time data integration capabilities as a core architectural requirement rather than an optional enhancement. Business processes increasingly depended on timely data synchronization across operational systems, analytical platforms, and customer facing applications. As organizations expanded their digital footprint, delays in data propagation translated directly into reduced responsiveness, inconsistent system behavior, and diminished decision making effectiveness. One primary driver was the growth of operational analytics and monitoring use cases. Business teams required dashboards and reports that reflected current system state rather than historical snapshots. In domains such as finance, logistics, and customer operations, metrics derived from stale data led to incorrect assessments and delayed intervention. Near real time integration enabled analytical platforms to consume continuously updated data streams, supporting more accurate situational awareness and faster operational response.

Another significant driver involved the rise of distributed application architectures. By 2019, many enterprises had adopted service oriented or microservices based designs where multiple applications depended on shared data entities. Maintaining consistency across these systems through synchronous database access introduced tight coupling and scalability constraints. Near real time integration provided an asynchronous alternative, allowing data changes to be propagated across services without direct dependency on the source system at request time. Regulatory and compliance requirements also contributed to demand for timely data integration. Industries subject to reporting and audit obligations increasingly required faster visibility into transactional activity to support risk management and regulatory oversight. Batch based reporting pipelines often failed to meet these expectations, particularly during periods of high

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

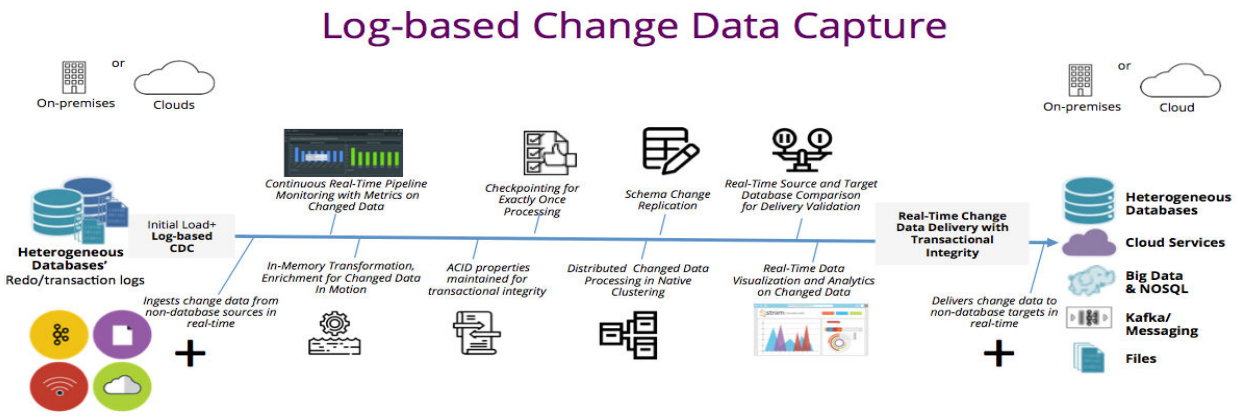
DOI:10.15662/IJRAI.2019.0201004

transaction volume. Continuous data integration improved the ability to detect anomalies, enforce controls, and produce accurate reports within tighter timeframes.

Hybrid architecture adoption further reinforced the need for near real time integration. As enterprises migrated analytics and integration workloads to cloud platforms while retaining core systems on premises, maintaining data coherence across environments became more complex. Latency introduced by batch transfers and manual synchronization processes undermined the benefits of cloud adoption. Near real time pipelines enabled a more seamless flow of data between environments, supporting hybrid operating models without excessive operational burden. Collectively, these drivers underscored the strategic importance of near real time data integration by early 2019. Enterprises sought solutions that could deliver continuous data movement with strong reliability, minimal source system impact, and manageable operational complexity. The following sections examine how Change Data Capture and streaming integration technologies addressed these drivers in practice.

III. ROLE OF CHANGE DATA CAPTURE IN NEAR REAL TIME INTEGRATION

Change Data Capture served as the foundational mechanism for enabling near real time data integration in enterprise environments by February 2019. Rather than relying on repeated queries or bulk extraction of source tables, CDC focused on observing and propagating incremental changes as they were committed in transactional systems. This approach aligned naturally with the requirements of continuous data movement, reducing latency while minimizing operational impact on production databases. Log based CDC had become the dominant technique for near real time integration due to its ability to capture changes directly from database transaction logs. By reading redo or write ahead logs, CDC processes could observe inserts, updates, and deletes after transaction commit without interfering with application execution paths. This preserved transactional ordering and atomicity, enabling downstream systems to apply changes consistently. Compared to trigger based or polling based approaches, log based CDC offered superior performance characteristics in high volume environments and reduced risk to source system stability.



Informatica PowerExchange was widely used in enterprise settings to implement log based CDC across a variety of relational database platforms. PowerExchange provided connectors that interpreted native database logs and translated committed changes into structured change records suitable for downstream consumption. Its integration with existing Informatica data integration tooling allowed organizations to incorporate CDC into established operational workflows, governance models, and monitoring practices. This made PowerExchange a practical choice for enterprises seeking to modernize data integration without replacing existing infrastructure wholesale. CDC also played a critical role in decoupling source systems from downstream consumers. By externalizing change capture from application logic, CDC eliminated the need for direct database access by consuming systems. This reduced tight coupling and limited the blast radius of downstream failures. Source systems could continue operating independently while CDC pipelines buffered and propagated changes asynchronously, improving overall system resilience.

Another important aspect of CDC in near real time integration involved managing data volume and change frequency. High transaction rates required CDC pipelines capable of sustaining continuous throughput without backlog

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

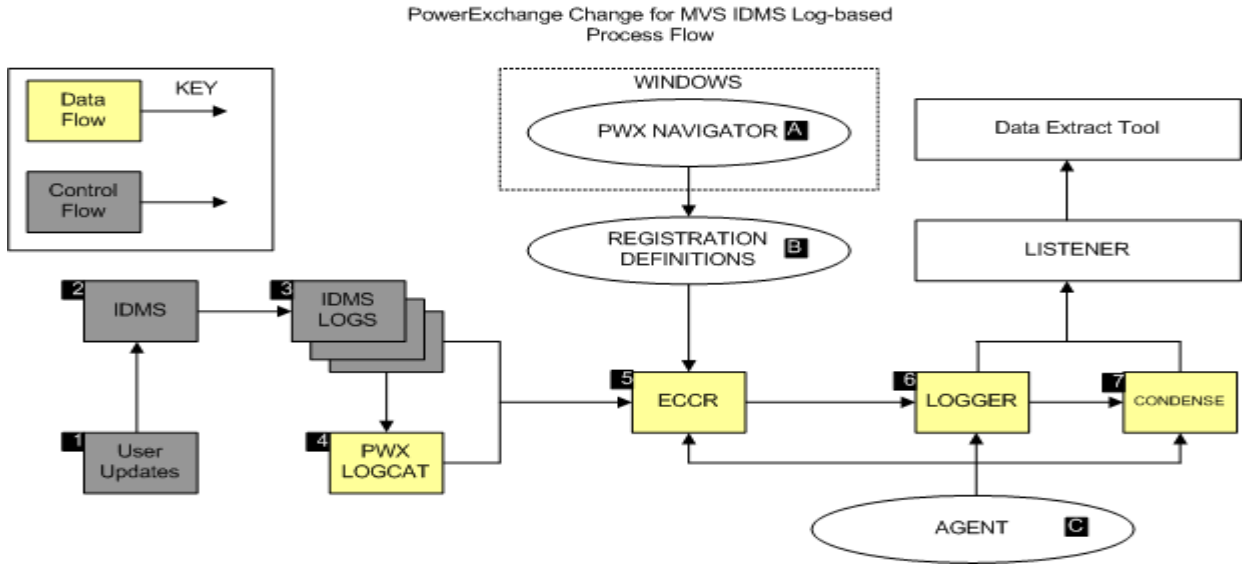
|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

accumulation. Log based CDC addressed this requirement by streaming changes incrementally rather than processing large datasets in discrete batches. This incremental model allowed integration pipelines to scale more predictably and reduced the need for costly backfill operations. Through these capabilities, Change Data Capture established itself as an essential component of near real time data integration architectures by early 2019. When combined with streaming ingestion and processing platforms, CDC enabled enterprises to construct reliable, low latency pipelines that bridged on premises and cloud environments. The next section examines how PowerExchange was architected and configured to support these integration patterns.

IV. POWEREXCHANGE ARCHITECTURE FOR LOG BASED CHANGE CAPTURE

By February 2019, Informatica PowerExchange had established itself as a mature and reliable platform for implementing log based Change Data Capture in enterprise environments. Its architecture was designed to integrate closely with transactional databases while minimizing impact on production workloads. PowerExchange operated by reading database transaction logs and extracting committed changes, transforming low level log records into structured change events that could be consumed by downstream integration processes. At the core of the PowerExchange architecture was the log reader component, which interfaced directly with database specific logging mechanisms. This component tracked log positions persistently, ensuring that changes were captured exactly once from the perspective of the source system. By maintaining checkpoints and restart positions, PowerExchange enabled CDC processes to recover from failures without data loss or duplication. This capability was essential for sustaining continuous capture in high volume environments where interruptions were inevitable.



PowerExchange introduced an abstraction layer that normalized change data across heterogeneous database platforms. Regardless of the underlying source system, captured changes were represented using consistent metadata structures that included operation type, primary key values, before and after images where applicable, and commit sequence information. This abstraction simplified downstream integration logic and allowed CDC pipelines to be reused across multiple source systems with minimal customization. Integration with Informatica data integration workflows allowed PowerExchange to function as part of a broader enterprise integration ecosystem. Captured change events could be routed into mappings, transformations, and delivery processes managed through Informatica tooling. This integration supported centralized configuration, security enforcement, and operational monitoring, aligning CDC processes with existing governance frameworks. For organizations already invested in Informatica platforms, this reduced the learning curve and operational friction associated with adopting CDC.

Operational considerations were integral to PowerExchange architecture design. Configuration options allowed tuning of capture latency, throughput, and resource utilization based on workload characteristics. Enterprises could balance near real time responsiveness against infrastructure constraints by adjusting commit intervals, buffering strategies, and extraction parallelism. These controls enabled PowerExchange to operate effectively across a wide range of

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

transactional volumes and system profiles. Through its log based architecture and integration capabilities, PowerExchange provided a robust foundation for near real time data capture by early 2019. Its ability to reliably extract and normalize transactional changes positioned it as a key component in enterprise integration pipelines, particularly when combined with cloud based streaming and ingestion services. The following section examines how Informatica Intelligent Cloud Services Streaming extended this foundation into cloud native, near real time integration workflows.

Component	Primary Responsibility	Architectural Role
PowerExchange	Log based change capture	Source system integration
CDC Log Reader	Read committed transactions	Preserve ordering and integrity
IICS Streaming	Continuous ingestion	Transport and buffering
Streaming Pipelines	Routing and light transforms	Near real time delivery
Cloud Targets	Analytics and applications	Data consumption

V. IICS STREAMING FOR CONTINUOUS INGESTION AND PROCESSING

By February 2019, Informatica Intelligent Cloud Services introduced streaming ingestion and processing capabilities that extended traditional integration workflows into near real time, cloud based execution models. IICS Streaming was designed to ingest continuous data flows, including CDC event streams, and deliver them to cloud targets with low latency and managed operational overhead. This capability aligned well with enterprise requirements for hybrid architectures, where transactional systems remained on premises while integration and analytics increasingly moved to the cloud. IICS Streaming functioned as an ingestion and routing layer that consumed event streams generated by CDC tools such as PowerExchange. Rather than relying on scheduled batch jobs, streaming pipelines operated continuously, processing incoming events as they arrived. This enabled downstream systems to receive updates shortly after source transactions were committed, supporting near real time synchronization without direct dependency on source databases. The streaming model reduced latency variability and provided a more predictable data flow pattern compared to batch based transfers.

A key architectural characteristic of IICS Streaming was its managed execution environment. By abstracting infrastructure provisioning, scaling, and fault handling, IICS reduced the operational complexity typically associated with running streaming systems. Enterprises could focus on defining ingestion logic, routing rules, and transformation steps without managing underlying cluster resources directly. This was particularly valuable for organizations transitioning from on premises integration platforms to cloud based services. IICS Streaming supported buffering and backpressure handling to accommodate variations in event volume and downstream processing capacity. When CDC event rates increased, the streaming layer absorbed bursts without overwhelming target systems. This buffering capability helped maintain stability across integration pipelines and reduced the likelihood of cascading failures during peak transaction periods. Combined with retention policies, it also enabled limited replay of events for recovery scenarios.

Integration with other Informatica services allowed IICS Streaming pipelines to participate in broader data integration and governance workflows. Streaming ingestion could feed data into cloud data warehouses, object storage, or downstream applications while leveraging consistent security policies and monitoring mechanisms. This unified approach simplified operational oversight and ensured that near real time pipelines adhered to enterprise standards for data handling. Through these capabilities, IICS Streaming complemented PowerExchange by providing a cloud native pathway for continuous data movement. Together, these technologies enabled enterprises to implement near real time data integration architectures that balanced low latency delivery with reliability and governance. The next section examines how PowerExchange and IICS Streaming were combined into end to end integration patterns suitable for enterprise deployment.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

VI. END TO END NEAR REAL TIME INTEGRATION PATTERNS USING POWEREXCHANGE AND IICS

By February 2019, enterprises combining PowerExchange with IICS Streaming began to converge on a set of repeatable architectural patterns for implementing near real time data integration across hybrid environments. These patterns reflected practical experience gained from operating CDC pipelines at scale and addressed common requirements related to latency, reliability, and operational governance. Rather than treating change capture and streaming ingestion as isolated components, successful implementations emphasized end to end flow design that aligned source system behavior with downstream consumption needs. A common pattern involved deploying PowerExchange close to the source database to perform log based change capture and publish normalized change records to an intermediate transport layer. These change records represented committed transactions and included sufficient metadata to support ordering, filtering, and downstream transformation. By localizing log access, this pattern minimized network overhead and reduced the risk of impacting transactional workloads. The captured changes were then forwarded to IICS Streaming through secure and reliable channels for continuous ingestion into cloud environments.

Within IICS Streaming, CDC events were typically routed through lightweight transformation and enrichment steps before delivery to target systems. These steps often included filtering irrelevant changes, mapping source fields to target schemas, and adding processing metadata such as ingestion timestamps. Keeping transformations minimal within the streaming layer helped preserve low latency characteristics while ensuring that downstream systems received data in usable formats. More complex transformations were deferred to downstream processing stages when necessary. Another widely adopted pattern focused on fan out distribution, where a single CDC stream served multiple downstream consumers. IICS Streaming enabled this by routing the same change events to different targets, such as cloud data warehouses, object storage, and operational applications. Each consumer processed the events independently, allowing different latency and consistency requirements to coexist without introducing tight coupling. This pattern reduced duplication of CDC extraction logic and simplified overall integration topology.

Checkpointing and replay capabilities formed an important part of end to end design. PowerExchange tracked log positions to ensure consistent capture, while IICS Streaming maintained offsets and processing state for streaming pipelines. Together, these mechanisms allowed integration flows to recover from transient failures and resume processing without data loss. Enterprises often defined clear recovery procedures that leveraged these checkpoints to minimize operational disruption. Through these integration patterns, enterprises were able to construct robust near real time pipelines that bridged on premises transactional systems and cloud based targets. The combination of PowerExchange and IICS Streaming provided a clear separation of responsibilities between change capture and event transport, enabling scalable and manageable integration architectures. The following section examines how these pipelines addressed data consistency and ordering requirements in distributed environments.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

V. DATA CONSISTENCY, ORDERING, AND LATENCY MANAGEMENT

Ensuring data consistency and correct ordering was a central concern in near real time integration pipelines built with PowerExchange and IICS Streaming. By February 2019, enterprises recognized that continuous data movement introduced new consistency challenges compared to batch based integration, particularly when changes were applied incrementally across distributed systems. Preserving transactional intent while maintaining acceptable latency required careful coordination between change capture, transport, and consumption layers. PowerExchange addressed consistency at the source by capturing only committed transactions from database logs and preserving commit order within the extracted change stream. This ensured that downstream systems observed changes in a sequence consistent with the source database state. Metadata such as commit timestamps and sequence identifiers enabled downstream pipelines to reason about ordering and apply changes deterministically. This capability was essential for maintaining correctness in systems that relied on incremental updates rather than periodic reconciliation.

Within IICS Streaming, ordering guarantees were maintained by processing events in defined streams and preserving sequence within those streams. While streaming pipelines allowed parallel processing for scalability, partitioning strategies were designed to ensure that related changes were processed in order. For example, events were commonly partitioned by primary key or logical business entity, allowing parallelism across entities while preserving order within each entity's change history. This approach balanced throughput with correctness in high volume scenarios. Latency management represented another critical aspect of near real time integration. Enterprises sought to minimize the time between source commit and downstream availability while avoiding instability caused by aggressive tuning. PowerExchange configuration options allowed adjustment of commit capture intervals and buffering behavior to control capture latency. Similarly, IICS Streaming pipelines could be tuned to balance processing efficiency against responsiveness, enabling predictable near real time delivery without excessive resource consumption.

Handling temporary backlogs and downstream slowdowns was essential for maintaining system stability. IICS Streaming provided buffering and flow control mechanisms that absorbed bursts of CDC events without overwhelming targets. During periods of sustained high load, latency increased gradually rather than resulting in pipeline failure. Monitoring lag metrics enabled operational teams to detect emerging bottlenecks and adjust capacity or processing logic accordingly. Through coordinated management of ordering, consistency, and latency, enterprises were able to operate near real time integration pipelines with confidence. These design practices ensured that continuous data movement did not compromise correctness or reliability, even as transaction volumes fluctuated. The next section examines governance, monitoring, and operational controls required to sustain these pipelines in production environments.

VI. GOVERNANCE, MONITORING, AND OPERATIONAL CONTROLS

By February 2019, enterprises operating near real time integration pipelines recognized that governance and operational controls were as critical to success as the underlying CDC and streaming technologies. Continuous data movement reduced tolerance for manual intervention and ad hoc troubleshooting, requiring integration pipelines to be designed with strong observability, predictable behavior, and alignment with enterprise governance standards. PowerExchange and IICS Streaming were typically deployed within controlled operational frameworks to ensure reliability at scale. Data governance considerations focused on ensuring that CDC streams adhered to enterprise policies for data quality, security, and compliance. Since PowerExchange captured transactional changes at a granular level, including potentially sensitive fields, organizations implemented filtering and masking rules to limit exposure of confidential data. These controls were often enforced early in the pipeline, either at capture or ingestion time, to reduce downstream risk. Alignment with enterprise data classification and access control policies ensured that near real time pipelines met regulatory and internal audit requirements.

Monitoring and observability were essential for managing continuous integration workflows. Unlike batch jobs with clear start and end points, streaming pipelines required continuous visibility into system health and performance. Operational teams monitored metrics such as CDC lag from source logs, streaming ingestion throughput, consumer offsets, and end to end latency. Alerts were configured to detect abnormal conditions, including stalled capture processes, growing backlogs, or repeated processing errors. These signals enabled proactive intervention before data freshness degraded significantly. Operational controls also addressed failure handling and recovery procedures. PowerExchange maintained persistent tracking of log positions to support restart without data loss, while IICS

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

Streaming preserved processing state to allow pipelines to resume from known offsets. Enterprises defined standardized recovery playbooks that outlined steps for restarting capture processes, validating downstream consistency, and resuming normal operation. These procedures reduced mean time to recovery and increased confidence in the resilience of near real time integration pipelines.

Concern	Control Mechanism	Risk Mitigated
Ordering	Commit sequence metadata	Out of order updates
Latency	Lag monitoring	Silent delays
Recovery	Log position checkpoints	Data loss
Backpressure	Streaming buffers	Target overload
Schema change	Versioned mappings	Pipeline breakage

Change management represented another important operational consideration. Modifications to source schemas, capture configurations, or streaming pipelines had a direct impact on data consumers. Enterprises adopted disciplined deployment practices, including testing changes in lower environments and coordinating releases across capture and ingestion components. Versioned configurations and rollback mechanisms helped mitigate risk during upgrades or schema evolution. Through robust governance, monitoring, and operational controls, enterprises were able to operate PowerExchange and IICS Streaming based integration pipelines reliably in production. These practices ensured that near real time data movement remained predictable, secure, and auditable despite the inherent complexity of continuous integration. The following section describes the methodology used to analyze these architectural patterns within the context of early 2019 enterprise environments.

VII. METHODOLOGY

This paper employs a qualitative architectural analysis methodology grounded in enterprise data integration practices as they existed by February 2019. Rather than relying on controlled experiments or benchmark driven comparisons, the methodology focuses on synthesizing documented industry implementations, vendor architecture guidance, and practitioner experience related to near real time Change Data Capture and streaming integration. This approach is appropriate for examining enterprise integration architectures that were widely deployed in practice but varied significantly based on organizational context and operational constraints. Primary sources for the analysis include Informatica product documentation, technical whitepapers, and case studies describing PowerExchange and IICS Streaming deployments in production environments. These materials provide insight into how enterprises configured CDC pipelines, managed hybrid connectivity, and addressed operational challenges such as latency, failure recovery, and governance. Emphasis is placed on sources that reflect real world operational behavior rather than purely conceptual designs.

The methodology involves decomposing near real time integration pipelines into functional layers, including change capture, event transport, streaming ingestion, transformation, and delivery to target systems. Each layer is examined in terms of its responsibilities, interaction patterns, and failure modes. By analyzing these layers individually and in combination, the paper identifies architectural patterns that enable reliable and scalable continuous data movement. Comparative analysis is also used to contrast near real time integration approaches with traditional batch based integration models. This comparison focuses on structural differences related to latency, coupling, scalability, and operational predictability. The goal is not to provide quantitative performance metrics, but to highlight architectural trade offs that influence system behavior under sustained transactional load.

To ensure temporal accuracy, the analysis is explicitly constrained to technologies, features, and operational practices that were available and commonly adopted by early 2019. Capabilities introduced in later platform versions or subsequent architectural trends are deliberately excluded to avoid retrospective bias. This constraint ensures that the findings accurately reflect the design decisions faced by enterprises during the period under study. Finally, the methodology prioritizes operational feasibility and governance alignment as evaluation criteria. Architectural patterns

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

are assessed based on their ability to support continuous throughput, preserve transactional consistency, and integrate with enterprise monitoring and control frameworks. This focus ensures that the analysis remains grounded in practical considerations relevant to production scale data integration.

VIII. FINDINGS AND OBSERVATIONS

The analysis shows that near real time data integration architectures built using PowerExchange and IICS Streaming provided a meaningful improvement over traditional batch based approaches in terms of latency, reliability, and operational flexibility. Enterprises adopting this combination were able to reduce data propagation delays from hours to minutes, and in many cases to near continuous delivery, without placing additional load on transactional source systems. This improvement enabled downstream platforms to operate on fresher data and supported more responsive operational and analytical use cases. One key finding is that log based CDC via PowerExchange proved highly effective in sustaining continuous change capture under high transaction volumes. By extracting changes directly from database logs, PowerExchange minimized contention with application workloads and preserved transactional ordering. Enterprises observed more predictable capture behavior compared to polling based approaches, particularly during peak transaction periods. This reliability at the capture layer was a prerequisite for achieving stable near real time pipelines end to end.

Another observation concerns the role of IICS Streaming in simplifying cloud side ingestion and routing. Managed streaming execution reduced the operational burden associated with provisioning and maintaining infrastructure, allowing teams to focus on pipeline design and data quality. The ability to buffer events and manage backpressure helped absorb fluctuations in change rates, preventing downstream overload and reducing the likelihood of pipeline failures. Enterprises benefited from clearer operational visibility into streaming throughput and lag. The findings also highlight the importance of clear separation of responsibilities between capture and ingestion layers. PowerExchange handled source specific concerns such as log interpretation and commit tracking, while IICS Streaming focused on transport, routing, and delivery. This separation improved modularity and made pipelines easier to evolve independently. Organizations that maintained this separation experienced fewer cascading failures and simpler troubleshooting.

Operational discipline emerged as a critical success factor. Enterprises that invested in monitoring lag metrics, defining recovery procedures, and coordinating schema changes across systems achieved more consistent near real time behavior. In contrast, pipelines lacking clear operational ownership or observability were prone to silent degradation, where latency increased gradually without immediate detection. Overall, the findings indicate that PowerExchange and IICS Streaming together formed a viable and practical foundation for near real time data integration by early 2019. When designed and operated with appropriate governance, these architectures supported continuous data movement at enterprise scale while maintaining reliability and transactional integrity.

IX. CHALLENGES AND LIMITATIONS

Despite their benefits, near real time integration pipelines using PowerExchange and IICS Streaming were not without challenges. One limitation involved the operational complexity associated with managing CDC at scale. PowerExchange required careful configuration and ongoing maintenance to handle database upgrades, log format changes, and evolving source schemas. Inadequate planning or monitoring at the capture layer could result in stalled pipelines or missed changes. Latency predictability also posed challenges in hybrid environments. Network variability between on premises systems and cloud services introduced fluctuations that were difficult to eliminate entirely. While IICS Streaming provided buffering and flow control, sustained spikes in transaction volume could still lead to increased lag. Enterprises often addressed this through conservative capacity planning and acceptance of bounded, rather than strict, latency guarantees.

Schema evolution remained a persistent source of risk. Changes to source database structures had direct impact on CDC streams and downstream consumers. Although PowerExchange captured structural changes reliably, coordinating updates across ingestion pipelines and target schemas required disciplined change management. Organizations lacking formal schema governance experienced higher rates of pipeline disruption during application releases. Cost considerations also influenced adoption. Near real time integration consumed continuous compute and network resources, which increased operational costs compared to batch processing. Enterprises had to balance the business

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

value of low latency data against infrastructure and licensing expenses, often limiting near real time pipelines to high value datasets.

Finally, organizational readiness affected the effectiveness of these architectures. Continuous pipelines required teams to adopt always on operational mindsets, with responsibility for monitoring and incident response extending beyond traditional batch windows. Enterprises that treated near real time integration as critical infrastructure, with clear ownership and support processes, were more successful than those that approached it as an incremental enhancement.

X. CONCLUSION

By February 2019, near real time data integration had become an essential capability for enterprises operating hybrid data architectures. The combination of Informatica PowerExchange for log based Change Data Capture and IICS Streaming for continuous ingestion provided a practical and enterprise ready approach to meeting these demands. Together, these technologies enabled incremental, low latency data movement while preserving transactional integrity and minimizing impact on source systems. This paper examined how PowerExchange and IICS Streaming were used to construct near real time integration pipelines, focusing on architectural patterns, consistency management, and operational controls. The analysis demonstrates that successful implementations relied on clear separation of concerns, disciplined monitoring, and alignment with enterprise governance frameworks. While not eliminating all sources of latency or complexity, these architectures represented a significant advancement over traditional batch based integration models.

The challenges identified in this study underscore that near real time integration is as much an operational and organizational endeavor as a technical one. Effective adoption required careful capacity planning, structured change management, and sustained operational oversight. Enterprises that addressed these dimensions were able to realize the benefits of continuous data movement while maintaining stability and predictability. In conclusion, PowerExchange and IICS Streaming based integration patterns as of early 2019 illustrate a mature stage in the evolution of enterprise data integration. They bridge the gap between legacy batch processing and fully event driven architectures, providing a reliable foundation for near real time data propagation in complex enterprise environments.

REFERENCES

1. Adiba Sabtu, Nur Fadzilah Mohd Azmi, Nor Nazihah A. Sjarif, Shafiza A. Ismail, Othman M. Yusop, Haslina Sarkan, Shahida Chuprat (2017). The Challenges of Extract, Transform and Loading (ETL) System Implementation for Near Real-Time Environment. 2017 International Conference on Research and Innovation in Information Systems (ICRIIS), 1-6. https://doi.org/10.1109/ICRIIS.2017.8002467
2. Rui J. Santos, Jorge Bernardino, Marco Vieira (2011). 24/7 Real-Time Data Warehousing: A Tool for Continuous Actionable Knowledge. 2011 IEEE 35th Annual Computer Software and Applications Conference (COMPSAC), 279-288. https://doi.org/10.1109/COMPSAC.2011.44
3. Srividya K. Bansal (2014). Towards a Semantic Extract-Transform-Load (ETL) Framework for Big Data Integration. 2014 IEEE International Congress on Big Data, 522-529. https://doi.org/10.1109/BigData.Congress.2014.82
4. Bhole Rahul Hiraman; Chapte Viresh M.; Karve Abhijeet C. (2018). A Study of Apache Kafka in Big Data Stream Processing. 2018 International Conference on Information, Communication, Engineering and Technology (ICICET), 1-4. https://doi.org/10.1109/ICICET.2018.8533771
5. Paul Le Noac'h; Alexandru Costan; Luc Bougé (2017). A Performance Evaluation of Apache Kafka in Support of Big Data Streaming Applications. 2017 IEEE International Conference on Big Data (Big Data), 4803-4806. https://doi.org/10.1109/BigData.2017.8258548
6. Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H. Luan, Hao Liang (2016). Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption. IEEE Internet of Things Journal, 3(6), 1171-1181. https://doi.org/10.1109/JIOT.2016.2565516
7. Avani Sharma; Tarun Goyal; Emmanuel S. Pilli; Arka P. Mazumdar; M. C. Govil; R.C. Joshi (2015). A Secure Hybrid Cloud Enabled Architecture for Internet of Things. 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 286-291. https://doi.org/10.1109/WF-IoT.2015.7389065

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

|| Volume 2, Issue 1, January-February 2019||

DOI:10.15662/IJRAI.2019.0201004

8. Sudhir Vishnubhatla. (2017). Migrating Legacy Information Management Systems to AWS and GCP: Challenges, Hybrid Strategies, and a Dual-Cloud Readiness Playbook. In International Journal of Scientific Research & Engineering Trends (Vol. 3, Number 6). Zenodo. https://doi.org/10.5281/zenodo.17298069
9. Hsinchun Chen, Roger H. L. Chiang, Veda C. Storey (2012). Business Intelligence and Analytics: From Big Data to Big Impact. MIS Quarterly, 36(4), 1165-1188. https://doi.org/10.2307/41703503
10. C. Mohan, Don Haderle, Bruce Lindsay, Hamid Pirahesh, Peter Schwarz (1992). ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging. ACM Transactions on Database Systems, 17(1), 94-162. https://doi.org/10.1145/128765.128770
11. Patrick Eugene O'Neil, Edward Cheng, Dieter Gawlick, Elizabeth O'Neil (1996). The Log-Structured Merge-Tree (LSM-Tree). Acta Informatica, 33(4), 351-385. https://doi.org/10.1007/s002360050048
12. Peter Bailis, Ali Ghodsi (2013). Eventual Consistency Today: Limitations, Extensions, and Beyond. Communications of the ACM, 56(5), 55-63. https://doi.org/10.1145/2460276.2462076
13. Werner Vogels (2009). Eventually Consistent. Communications of the ACM, 52(1), 40-44. https://doi.org/10.1145/1435417.1435432
14. Seth Gilbert, Nancy Lynch (2002). Brewer's Conjecture and the Feasibility of Consistent, Available, Partition-Tolerant Web Services. ACM SIGACT News, 33(2), 51-59. https://doi.org/10.1145/564585.564601
15. Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, Ion Stoica (2013). Discretized Streams: Fault-Tolerant Streaming Computation at Scale. SOSP '13: Proceedings of the 24th ACM Symposium on Operating Systems Principles, 423-438. https://doi.org/10.1145/2517349.2522737
16. Sudhir Vishnubhatla. (2018). From Risk Principles to Runtime Defenses: Security and Governance Frameworks for Big Data in Finance. In International Journal of Science, Engineering and Technology (Vol. 6, Number 1). Zenodo. https://doi.org/10.5281/zenodo.17452405