American Journal of Advanced Technology and Engineering Solutions

Volume: 2; **Issue:** 4
Pages: 217-254
eISSN: 3067-5146

open access

Crossref

PEER-REVIEWED

Advanced Computing Frameworks for Real-Time SAP S/4HANA Retail Business Intelligence: Optimizing Data Processing, Latency, and System Reliability

Zakia Afroz[1]; Khairum Nahar Pinky[2]

[1]. Assistant Manager, British American Resource Center (BARC), Dhaka, Bangladesh.
Email: juthybd61@gmail.com

[2]. Data Analyst, ISHO Limited, Bangladesh.
Email: khairumnahar@gmail.com

Doi: 10.63125/xk5j7g56

Received: 09 September 2022; **Revised:** 10 October 2022; **Accepted:** 12 November 2022; **Published:** 08 December 2022

Abstract

This study addresses the persistent problem that SAP S/4HANA retail business intelligence environments often struggle to achieve simultaneously low latency, high data freshness, and dependable reliability at enterprise scale, particularly when cloud and hybrid deployments introduce additional integration and operational complexity. The purpose was to quantify which advanced computing framework combinations are most consistently associated with improved real-time BI outcomes in retail decision workflows. Using a quantitative cross-sectional, case-based design, the study coded a sample of 60 eligible enterprise and cloud-oriented cases from the reviewed corpus (N = 60). Key variables included framework adoption categories (for example CDC or incremental refresh, in-memory or HTAP, streaming, distributed processing, cloud-native orchestration), outcome constructs measured via a 5-point Likert evidence scale (Latency Improvement Evidence LIE, Processing Efficiency Evidence PEE, Reliability and Continuity Evidence RCE), and workflow clusters (inventory, promotions, fulfillment, anomaly detection). The analysis plan applied frequency distributions and descriptive statistics, then conducted evidence-based hypothesis aggregation using a predefined support rule (at least 60% directional support and mean evidence score at or above 3.50). Findings show strong concentration of CDC or incremental refresh integration at 73.3% (44/60), in-memory or HTAP execution at 68.3% (41/60), streaming at 60.0% (36/60), cloud-native orchestration at 51.7% (31/60), and distributed processing at 48.3% (29/60). Overall evidence strength was high for latency and processing (LIE M = 3.84, SD = 0.71; PEE M = 3.76, SD = 0.69) with moderate-strong reliability (RCE M = 3.62, SD = 0.74). Headline results supported all three hypotheses: hybrid multi-layer stacks outperformed single-layer designs for latency (77.8% supportive, 28/36; hybrid LIE M = 4.08 vs 3.41), CDC and event-driven strategies outperformed batch refresh for freshness and reporting delay (CDC subgroup LIE M = 3.97 vs 3.22), and observability plus automated failover improved reliability versus monitoring-only patterns (high RCE 74.1% vs 42.1%; RCE M = 3.98 vs 3.28). Implications suggest practitioners should prioritize CDC-driven ingestion and incremental maintenance, adopt hybrid stacks that combine in-memory, streaming, and distributed compute, and operationalize reliability through observability and automated recovery, while tracking end-to-end latency as a decomposed pipeline metric aligned to core retail workflows where inventory (30.0%) and promotion monitoring (26.7%) dominate use cases.

Keywords

SAP S/4HANA, Real-Time Business Intelligence, Change Data Capture, In-Memory Analytics, Observability and Failover;

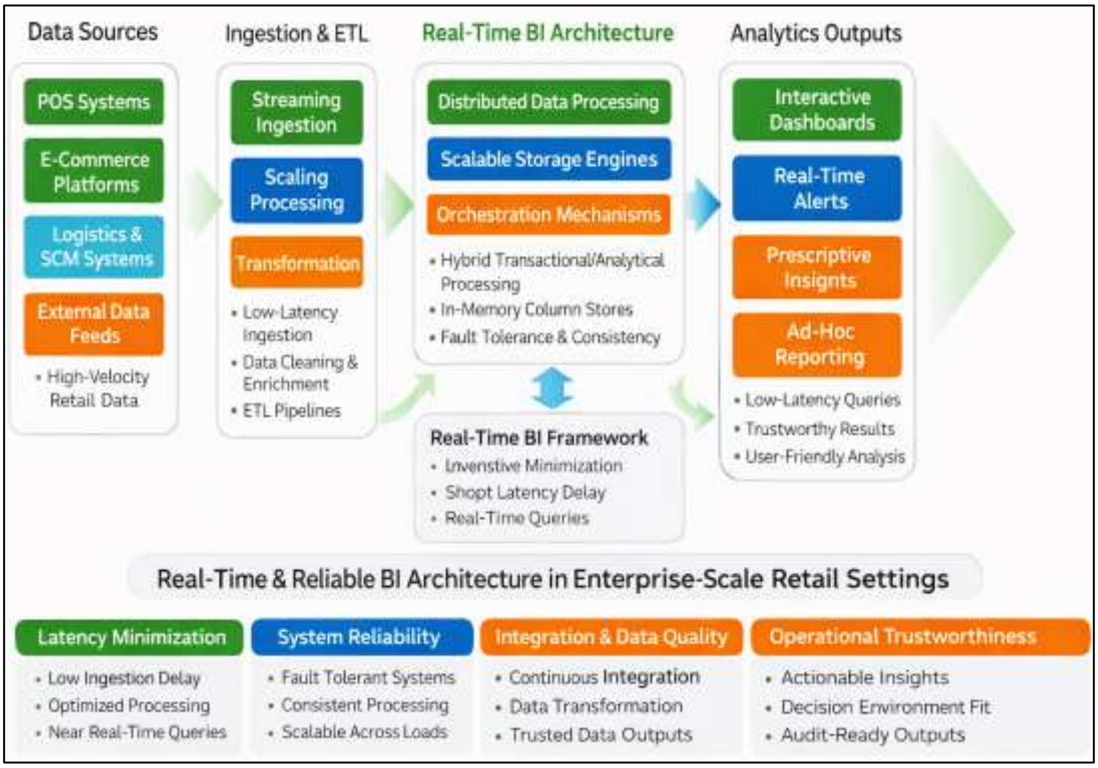
INTRODUCTION

Business intelligence (BI) in retail business operations is commonly defined as the set of architectures, processes, and analytical practices that transform operational and external data into timely, decision-relevant information products such as reports, dashboards, alerts, and prescriptive insights (Abadi et al., 2005). In the information-systems literature, BI is treated as an organizational capability that combines data quality, integration, and user access to support managerial decision environments and measurable success outcomes (Akidau et al., 2015). In accounting and performance-focused streams, BI is framed as an IT-enabled measurement and analysis capability that links business process performance to organizational performance through information-rich process visibility (Abadi et al., 2008). These definitions align with enterprise-scale retail settings where transaction intensity, assortment complexity, and multi-channel customer interactions produce high-velocity operational data that must be converted into low-latency intelligence artifacts that remain consistent with governance and controllership requirements (Armbrust et al., 2015). Within global retail, BI significance is amplified by cross-border supply networks, heterogeneous point-of-sale ecosystems, and regionally regulated data practices that require both technical performance and operational trustworthiness in analytic outputs (Biswas et al., 2020). The "real-time" qualifier in BI denotes the operational objective that analytic views and signals are refreshed at a cadence aligned with business action cycles, often approaching event-time immediacy for pricing, inventory, fraud, fulfillment, and promotion control loops (Brewer, 2012). Achieving this objective depends on advanced computing frameworks that integrate streaming ingestion, scalable processing, and transactional consistency while maintaining system reliability under bursty loads (Malewicz et al., 2010). Research on decision-quality pathways has shown that BI's contribution is mediated through management quality and information handling that shape the usefulness and credibility of analytic outputs. In international retail enterprises, the need for low-latency and reliable BI is linked to competitive operational execution and service-level commitments across geographically distributed stores and logistics nodes, making computing architecture a core determinant of BI feasibility and quality (Kemper & Neumann, 2011).

Advanced computing frameworks for real-time BI are typically constructed from distributed data-processing models, scalable storage engines, and orchestration mechanisms that coordinate ingestion, transformation, and query execution (Melnik et al., 2010). Foundational distributed-processing work formalized the principles of large-scale batch transformation with fault tolerance and locality-aware execution, and subsequent dataflow abstractions extended these principles toward unified models that support streaming and batch semantics for continuous analytics (Akidau et al., 2013). Retail BI workloads also require flexible high-level languages and compilation strategies to express data transformations across diverse sources, which is reflected in research on dataflow scripting languages designed for large datasets. Stream-processing research established architectural patterns for continuous queries, stateful operators, and backpressure-aware execution that enable near-real-time monitoring and alerting across event streams. Interactive analysis systems for web-scale datasets further developed execution strategies for low-latency aggregation and drill-down over nested and columnar representations (Gupta & George, 2016). Graph-based analytics, used in retail for recommendation affinity and fraud rings, has been supported by large-scale graph-processing frameworks that standardize vertex-centric computation across distributed clusters (Corbett et al., 2013). These computing models address key retail BI requirements: ingestion of heterogeneous events (store transactions, e-commerce clicks, shipment scans), transformation into conformed analytical structures, and interactive querying that supports operational decisions within strict time windows. International scale introduces additional constraints such as variable network conditions, multi-region deployments, and localization requirements that shape both performance and reliability expectations (Färber et al., 2012). BI success research ties these technical capabilities to organizational outcomes by emphasizing the roles of integration and data quality in satisfying user and decision needs (Plattner, 2009). Thus, advanced computing frameworks are not only "faster engines"; they are socio-technical assemblies where processing models, storage semantics, and operational governance collectively determine whether real-time retail BI outputs are trusted, timely, and consistently reproducible (Mikalef et al., 2018).

A central technical concept for this study is latency, commonly defined as the elapsed time between data generation (or event occurrence) and the availability of a correct, consumable analytic result. In real-time BI, latency is multidimensional and includes ingestion delay, processing time, queuing, synchronization, and query response time, with tail latency often determining user experience and operational viability (Shute et al., 2013). Large-scale service research has shown that tail latency characteristics can dominate user-visible responsiveness and system-level efficiency in distributed environments (Ghazanfari et al., 2011). Latency minimization in BI is also shaped by data layout and execution strategies; columnar storage has been empirically associated with improved analytic scan efficiency through reduced I/O and better compression, which directly influences query-time latency for aggregation-heavy workloads (Dean & Barroso, 2013). The rise of hybrid transactional/analytical processing (HTAP) further reframed latency by aiming to run operational transactions and analytics on the same dataset without extract-and-load delays (Isik et al., 2011).

Figure 1: Real-Time Retail BI Architecture for Low-Latency and Trustworthy Analytics Outputs



Early in-memory approaches proposed a common database design for OLTP and OLAP on in-memory column stores, introducing a path toward reduced synchronization and reduced warehouse refresh latency. Hybrid main-memory systems extended this idea by enabling analytical queries on transaction-consistent snapshots created efficiently at runtime (Olston et al., 2008). Transaction-engine advances for in-memory OLTP, exemplified by high-performance in-memory architectures, contributed mechanisms that reduce contention and improve predictable response times under high concurrency. Retail BI latency also includes the “organizational latency” of producing decision-ready intelligence, connecting technical delays to information logistics and system capability maturity (Thomson et al., 2012). Empirical BI research has operationalized quality of decision making as a measurable construct influenced by BI management, reinforcing that low latency is meaningful when analytic outputs remain accurate, comprehensible, and aligned with decision contexts. Hence, the literature supports treating latency as an integrated property of compute, storage, and governance pipelines rather than a single query-time metric (Torres et al., 2018). Reliability and system trustworthiness form the second core concept in the title, and they are commonly defined through the system’s ability to deliver correct results consistently under failures, load spikes, and operational changes (Verma et al., 2015). Distributed-systems literature characterizes reliability through fault tolerance, replication, consistency guarantees, and operational resilience. Key-value storage research demonstrated that high availability and partition tolerance can be achieved through decentralized replication and failure handling, shaping how modern data platforms balance durability and responsiveness (Vogels, 2009). Consistency research formalized eventual consistency as a practical model for distributed data services, clarifying how systems can provide usable semantics under network variability and replica divergence (DeCandia et al., 2007). At the same time, distributed-database research underscored the importance of globally consistent transactions for correctness and auditability, particularly relevant to retail financial reconciliations and controlled reporting. Globally distributed transactional systems provided designs for external consistency and high availability across regions, creating a foundation for trustworthy multi-site data processing (Popovič et al., 2012). Complementary research on large-scale production databases showed how structured schemas, distributed transactions, and operational tooling support correctness at scale in real business services (Wamba et al., 2017). In streaming contexts, reliability includes exactly-once or effectively-once processing, durable state management, and replay mechanisms (Diaconu et al., 2013). Stream-processing work at internet scale introduced durable state and event-time coordination approaches that maintain correctness under failures and reprocessing (Mikalef et al., 2019). Operational orchestration research has also demonstrated that cluster management and scheduling decisions influence reliability, availability, and predictability of performance across multi-tenant workloads. The reliability concept is tightly linked to BI success constructs: user satisfaction, decision support quality, and organizational culture for analytical decision making have been associated with realized BI success. Therefore, the literature positions system reliability as both a technical requirement and a determinant of whether BI outputs are accepted as credible inputs to retail decision processes (Wieder & Ossimitz, 2015).

In enterprise retail contexts, SAP S/4HANA Business Intelligence is frequently discussed as an embedded or tightly integrated analytics approach where transactional and analytical views can share a common data foundation, reducing data duplication and refresh delays. The underlying in-memory data management paradigm in SAP HANA has been described in database research as a modern architecture that leverages columnar storage, compression, and in-memory execution for mixed workloads, supporting low-latency analytic operations close to transactional data (Elbashir et al., 2008). Work on SAP HANA transaction processing clarified mechanisms that address transactional performance requirements within a column-store-oriented design, contributing to the feasibility of operational BI without separate warehouse staging for many use cases (Isik et al., 2013). This aligns with broader HTAP research that describes how snapshot isolation, concurrency control, and memory-resident execution can make real-time BI queries feasible alongside OLTP workloads (Jia et al., 2013). These architectures complement column-store findings about analytic efficiency and support a retail BI pattern where pricing dashboards, inventory heatmaps, promotion effectiveness, and store performance indicators are computed with reduced pipeline delays. However, the international retail setting introduces integration realities such as multiple source systems, data quality variability, and region-specific operational definitions that preserve the relevance of data warehousing and information logistics practices. Real-time warehousing research and implementation studies emphasize structured screening, integration, and transformation methods to maintain interpretability and comparability of analytical indicators across organizational units (Sikka et al., 2012). BI evaluation frameworks for enterprise systems provide structured criteria to assess BI capability and readiness, supporting cross-sectional comparisons in case-based studies (Krishnamoorthi & Mathew, 2010). Accordingly, SAP S/4HANA real-time BI optimization can be positioned as a synthesis problem: aligning in-memory transactional analytics, streaming/ETL practices, and reliability engineering to achieve low-latency and trustworthy intelligence outputs that remain consistent with enterprise control needs (Dean & Ghemawat, 2008).

This study is designed to examine, organize, and synthesize scholarly and practice-oriented evidence on advanced computing frameworks that enable real-time SAP S/4HANA retail business intelligence with explicit emphasis on optimizing data processing efficiency, reducing end-to-end latency, and strengthening system reliability. The first objective is to identify and classify the major framework categories used in SAP-centered retail BI ecosystems, including in-memory transactional analytics, distributed and parallel processing models, event-driven and change-data-capture pipelines, stream processing engines, hybrid transactional/analytical architectures, and cloud-native orchestration layers that coordinate compute and storage resources. The second objective is to analyze how these frameworks are applied across the full BI pipeline—from source event generation and data movement through transformation, semantic modeling, query execution, and visualization refresh—so that latency is treated as a measurable, multi-stage performance property rather than a single system characteristic. A third objective is to evaluate the dominant technical strategies reported for latency control and processing optimization, such as pushdown computation, incremental aggregation, workload isolation, partitioning, caching, snapshot-based execution, and resource elasticity, and to organize them into comparable theme groups that can be mapped to retail decision scenarios. The fourth objective is to synthesize evidence on reliability-oriented engineering practices that accompany real-time BI implementations, including high availability and disaster recovery patterns, fault tolerance mechanisms, operational monitoring and observability design, automated recovery workflows, and governance controls that protect correctness and continuity under peak demand and component failures. A fifth objective is to develop a case-study–based cross-sectional comparison structure that links framework combinations to typical retail use cases such as promotion performance monitoring, inventory availability analytics, pricing intelligence, omnichannel fulfillment visibility, and anomaly detection in transactions, enabling consistent comparison of architectural choices under different business criticality levels. Finally, the study aims to produce a structured evidence map that supports objective-driven assessment of proposed hypotheses by translating qualitative findings into light quantitative summaries such as frequency counts, cross-tabulated patterns, and coded strength-of-evidence indicators. Through these objectives, the introduction positions the research as a systematic synthesis effort that clarifies what framework configurations are most commonly reported, how they are justified in terms of processing and latency performance, and how reliability is engineered and documented within real-time SAP S/4HANA retail BI environments.

LITERATURE REVIEW

The literature on advanced computing frameworks for real-time SAP S/4HANA retail business intelligence spans multiple research streams that collectively explain how organizations engineer low-latency, high-throughput, and reliable analytics on top of enterprise resource planning ecosystems. At one level, business intelligence scholarship frames BI as a decision-support capability shaped by data integration, information quality, governance, and user-facing analytics artifacts, while retail-oriented studies emphasize the continuous nature of operational decisions and the need for timely insights across pricing, inventory, promotions, and fulfillment. At another level, database and distributed-systems research contributes the technical foundations for real-time analytics, including in-memory and columnar processing for rapid aggregation, hybrid transactional/analytical processing designs that reduce delays between operational events and analytical visibility, and distributed execution models that scale transformations and queries across large datasets. Complementary work on data integration and pipeline engineering examines how architectural choices such as batch ETL, ELT, change data capture, and event-driven streaming influence end-to-end freshness and the stability of analytical metrics. A further body of research addresses reliability and resilience as first-class requirements for analytics services, emphasizing fault tolerance, high availability, replication strategies, tail-latency control, and operational observability practices that keep BI responsive under peak demand and partial failures. Within SAP-centered environments, studies and technical analyses describe how S/4HANA and HANA-based data management can support embedded analytics and operational reporting, while also identifying integration realities that continue to motivate external analytical stores, semantic layers, and multi-system harmonization in complex retail enterprises. Across these strands, the literature reveals that "real-time BI" is not defined by a single technology but by coordinated design decisions across ingestion, processing, storage, serving, and governance layers, each contributing to measurable latency and reliability outcomes. This review therefore treats advanced computing frameworks as an interlocking set of mechanisms that enable fast analytics while preserving correctness, interpretability, and service continuity, and it structures prior evidence around framework categories, pipeline optimization themes, and resilience patterns relevant to international retail operations running SAP S/4HANA.

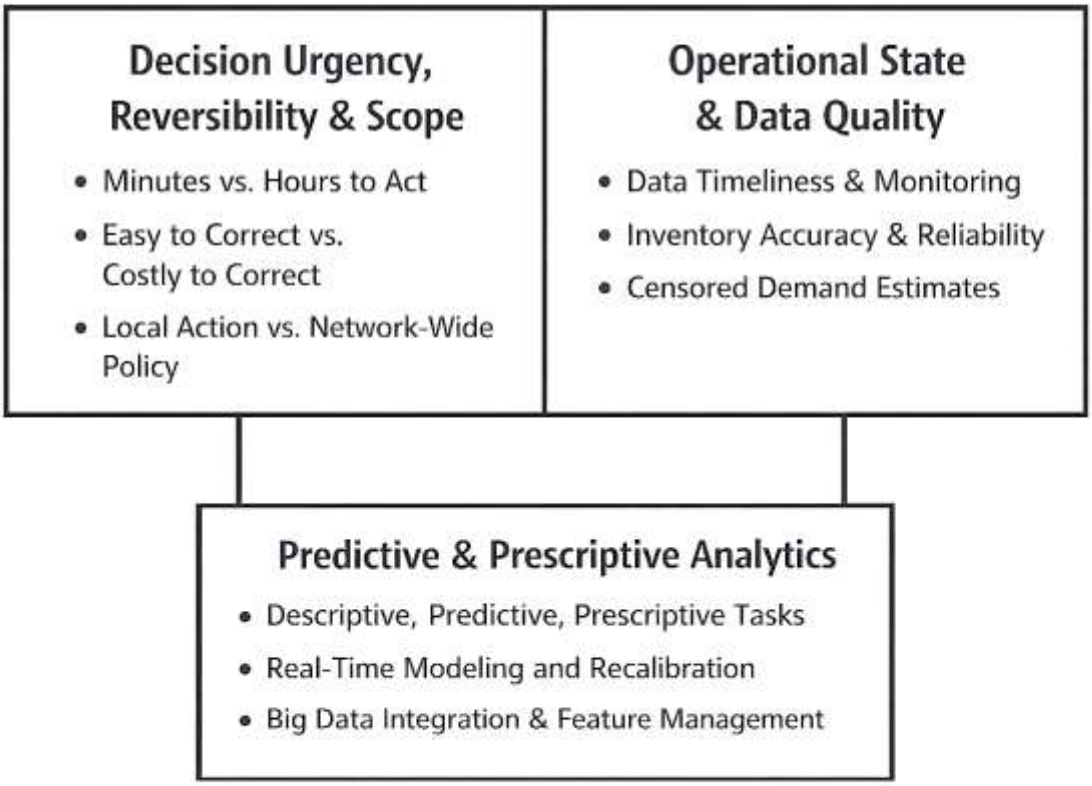
Real-Time Retail BI Requirements and Decision Workflows

Real-time retail business intelligence is commonly framed as an operational response to shoppers who move fluidly across touchpoints and expect consistent availability, pricing, and service promises. As retailers expand from multi-channel coordination toward omni-channel integration, the analytical unit of work shifts from isolated store or web transactions to continuous customer–product–fulfilment journeys that must be observed and reconciled at high frequency (Verhoef et al., 2015). This shift increases the need for a disciplined requirements view that separates decision urgency (minutes versus hours), decision reversibility (easy to correct versus costly to unwind), and decision scope (local store action versus network-wide policy). Within that requirements view, real-time BI is less about a single dashboard refresh rate and more about orchestrating a chain of time-sensitive tasks: capture events, validate them, attach business meaning, compute signals, and route those signals to the people or rules that act. For retail executives, the most visible decisions include markdown and promotion steering, substitution rules for out-of-stocks, and fulfillment rebalancing across stores and distribution nodes. For planners and store operators, the daily reality is exception management: identifying fast-moving items, reconciling mismatched on-hand counts, and detecting process breakdowns before they propagate into cancellations or lost sales. Omni-channel logistics research shows that fulfilment choices and last-mile design parameters create distinct information needs, because picking models, delivery modes, and customer handoff options change where latency accumulates and where failures occur (Hübner et al., 2016). Accordingly, literature treats decision workflows as coupled loops that connect demand signals, inventory states, and capacity constraints, requiring analytics that are fast enough to preserve optionality and stable enough to avoid oscillating decisions under noisy data. In SAP-centered environments, these workflow requirements are often expressed as near-real-time visibility for order-to-cash, procure-to-pay, and store replenishment processes, where operational BI must align with master data, pricing conditions, and authorization boundaries.

A second stream emphasizes that retail decision workflows depend on the quality and observability of operational state, especially demand that is partially observed and inventory that is imperfectly recorded. Store sales data are censored by stock availability, so the observed point-of-sale history can understate true demand when shelves empty or when substitutions occur, which complicates both rapid replenishment and the interpretation of BI alerts. Operational-visibility research reviews how censored demand and inventory record inaccuracy jointly erode the decision maker's situational awareness, and it explains why analytics must sometimes infer missing information rather than simply report it (Chen & Mersereau, 2015). From a computing-framework perspective, this literature is important because it clarifies what "real time" means for the business: a fast pipeline that propagates wrong inventory states can accelerate error, while a slightly slower pipeline that improves state accuracy can stabilize downstream decisions. In this view, reliability for real-time BI includes epistemic reliability, meaning the system provides signals that are consistently interpretable under known data imperfections, not only infrastructure reliability such as uptime. Methodologically, the workflow problem often reduces to estimating latent demand and latent inventory while transactions continue to arrive, which motivates incremental estimation and continuous reconciliation as first-class workload types. Empirical operations research demonstrates that when inventory uncertainty is ignored in censored-demand estimation, demand can be systematically biased downward, and that a practical correction can be built from an error statistic that can be learned from routine records (Mersereau, 2015). These findings connect to SAP S/4HANA retail BI because many metrics—fill rate, on-shelf availability, order promise accuracy, and substitution rates—are functions of both event timeliness and state fidelity. Consequently, literature positions real-time BI frameworks as controls that manage a trade space among speed, accuracy, and governance: faster computation tightens feedback loops, while validation and exception handling protect decision correctness.

A third body of work links real-time retail BI requirements to the analytic tasks that managers expect the system to support, ranging from descriptive monitoring to predictive and prescriptive interventions. Retail organizations rarely treat streaming signals as ends in themselves; they treat them as inputs to decisions about assortment, pricing, promotions, staffing, and replenishment that must be made under uncertainty and frequently revised. In this context, advanced computing frameworks are evaluated by their ability to combine heterogeneous data streams with domain theory and modeling so that rapid computation yields actionable signals rather than noise. Research on big data and predictive analytics in retailing highlights that value creation depends on mixing new data sources with principled modeling choices, including experimentation, causal inference logic, and data-compression strategies that make high-volume inputs tractable for decision cycles (Bradlow et al., 2017).

Figure 2: Real-Time Retail BI Requirements and Decision Workflows

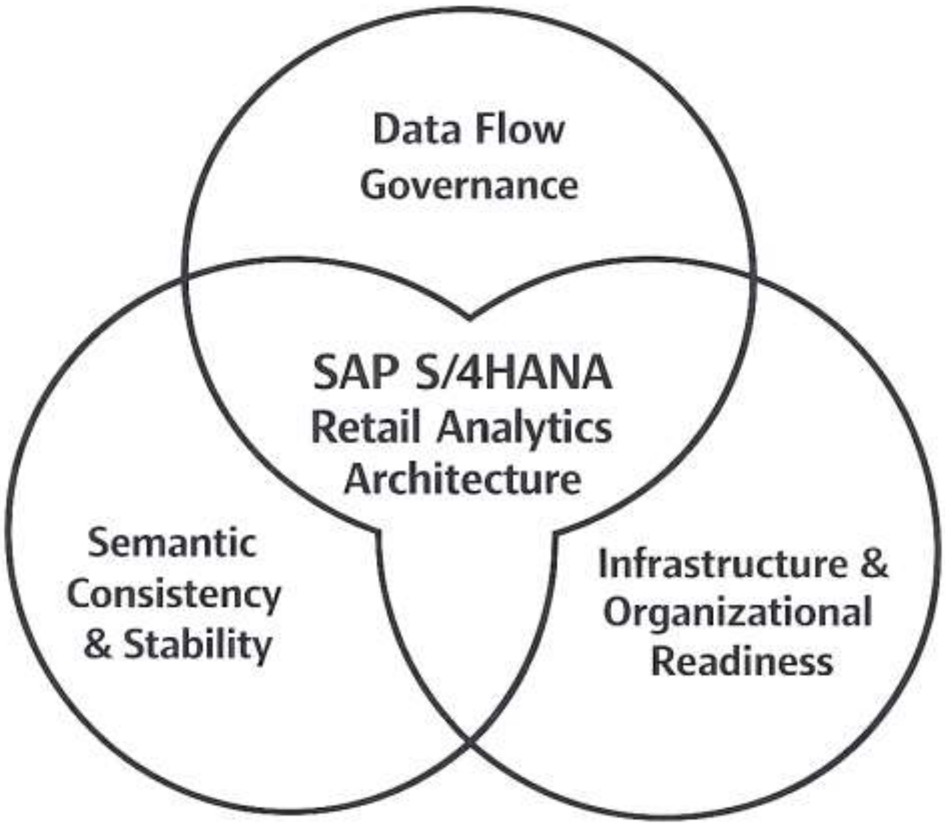


These requirements imply that real-time BI workflows must incorporate model training and scoring as operational workloads, not occasional offline projects, because the metrics displayed to executives often require continuous feature updates and recalibration. At the workflow level, this turns the BI pipeline into a closed-loop system: events generate indicators, indicators trigger actions, actions create new events, and the system must maintain lineage so that analysts can attribute performance changes to interventions. For SAP S/4HANA retail deployments, the closed-loop requirement is reflected in the need to align predictive outputs with transactional objects such as articles, sites, promotions, and orders, and to expose assumptions through layers that business users trust. It also elevates latency to a multi-criteria constraint: some tasks prioritize freshness (minute-level exception alerts), some prioritize consistency (financial and margin reporting), and some prioritize stability (experiment readouts and uplift estimates). Therefore, literature motivates real-time BI architectures that support differentiated service levels across workloads, enabling concurrent monitoring, learning, and optimization while preserving governance and auditability.

SAP S/4HANA Retail Analytics Architecture and Data Flow

SAP S/4HANA retail analytics architecture can be described as an ERP-centered data ecosystem in which transactional execution, semantic modeling, and analytical consumption are coordinated to deliver business intelligence outputs with minimal handoffs and controlled meaning. At the core, retail processes generate operational objects—sales documents, goods movements, pricing conditions, inventory valuations, and customer fulfillment events—that must be transformed into analyzable measures and dimensions without breaking the accounting and master-data logic embedded in the ERP. In this architecture, the "data flow" is not only a movement of records; it is also the movement of business definitions, because KPIs such as net sales, gross margin, stock coverage, and sell-through depend on consistent mappings of product hierarchies, site structures, calendars, and valuation rules. The literature on BI–ERP integration emphasizes that organizations struggle when ERP and BI are treated as separate programs, because integration decisions determine whether analytics aligns with operational truth or becomes a parallel narrative of the business (Nofal & Yusof, 2013). Within S/4HANA landscapes, this issue is expressed through the choice of how operational data is exposed for analytics: directly from live transactional tables, through curated semantic views, through replicated staging layers, or through enterprise data warehouse hubs that standardize and historize facts. Retail organizations typically need all of these patterns in combination, because operational reporting may require immediate visibility for store replenishment and exception management, while strategic analytics may require historized, conformed data integrated with external signals. Architecture therefore acts as a governance mechanism that controls which data flows are "authoritative," which are "situational," and which are "exploratory," and it also determines the latency boundaries and failure domains that shape end-user trust. A requirements-consistent architecture must also define where transformations occur, which systems own derived measures, how lineage is recorded, and how analytical authorization mirrors operational segregation of duties, making data flow inseparable from enterprise design choices.

Figure 3: Sap S4hana Retail Analytics Architecture And Data Flow Framework



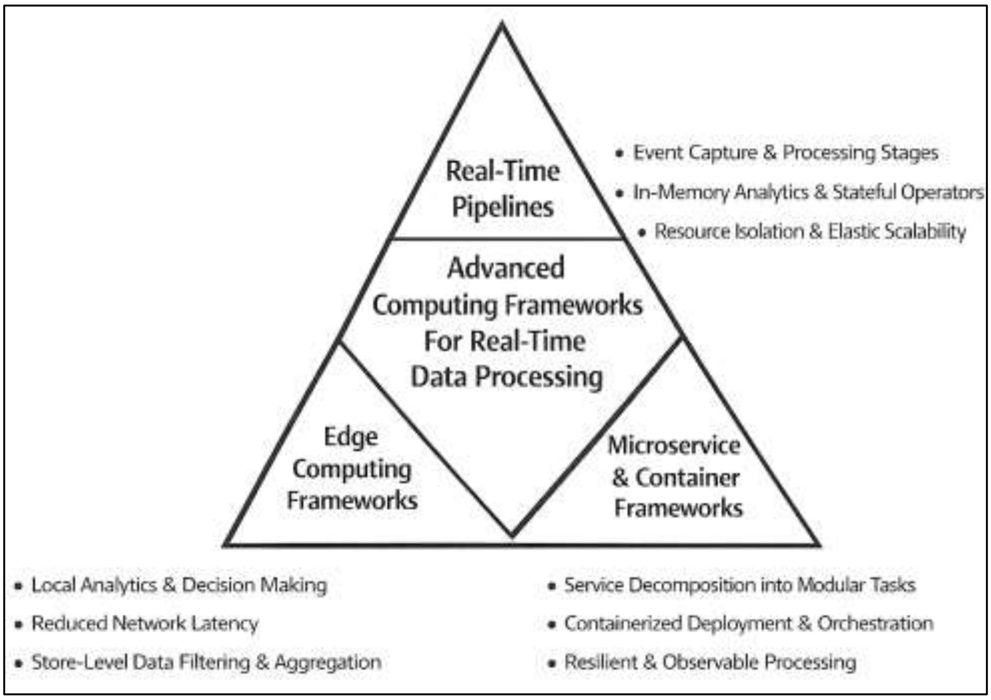
A second defining aspect of S/4HANA retail analytics architecture is the layered representation of meaning between raw operational structures and business-facing consumption. Retail BI requires a semantic layer that stabilizes definitions across departments and countries, because even small inconsistencies in pricing logic, unit conversions, or time-bucketing can produce conflicting KPI narratives. Research on enterprise data integration highlights that integration challenges are partly technical and partly semiotic: data sources may be syntactically connectable while still carrying incompatible meanings, quality assumptions, and contextual interpretations (Gailly et al., 2014). This perspective is directly relevant to S/4HANA environments because ERP data models encode business context through document types, status codes, valuation classes, and workflow states that are not automatically intelligible to downstream analytics tools. The architecture therefore depends on deliberate modeling choices that translate operational context into analytic structures that remain summarizable and aggregation-safe for BI users. Work on semantic assessment of summarizability in self-service BI reinforces that aggregation correctness is a core quality property in analytical models, because business users require trustworthy rollups along product, customer, and location dimensions without hidden double counting or broken grain (Ibáñez et al., 2017). In retail, summarizability failures frequently appear when omni-channel orders create multiple fulfillment steps, when returns cross channels, or when promotions create overlapping conditions; analytic models must therefore define grain, keys, and aggregation rules that preserve interpretability. Architecture and data flow decisions also determine how self-service exploration is bounded: organizations may allow flexible slicing while protecting validated definitions for finance-sensitive measures. As a result, the "virtual data model" concept commonly used in ERP-centered analytics can be interpreted as a governance-driven semantic contract that controls how operational data becomes BI-ready, making semantic stability a central design goal alongside speed.

A third architectural dimension concerns the infrastructural and organizational conditions that enable real-time behavior across the end-to-end pipeline. Even when S/4HANA provides fast access to operational facts, retail BI performance depends on the surrounding data infrastructure: integration services, staging rules, compute resources, and the operational discipline used to monitor and maintain them. Empirical work examining the integration of BI into ERP environments suggests that data infrastructure and resource support influence BI–ERP integration outcomes, highlighting the practical importance of warehousing components, OLAP structures, vendor choices, and operational readiness as enabling conditions (Chou et al., 2015). This evidence aligns with the observation that retail BI architectures are typically hybrid: they combine embedded analytics for operational monitoring with enterprise or departmental warehousing patterns for harmonization, history, and cross-domain integration. The data flow in such hybrids is often governed by enterprise architecture decisions that define integration boundaries, ownership, and standardization, because real-time BI workloads create cross-team dependencies among finance, supply chain, store operations, and IT. A systematic mapping study on enterprise architecture in enterprise integration emphasizes that integration is a continuous enterprise concern rather than a one-time technical project, and that architecture functions as the coordinating logic that aligns systems, processes, and governance across complex integration landscapes (Banaeianjahromi & Smolander, 2016). In SAP-centered retail BI, this implies that low latency is achieved not only by faster databases or stronger compute frameworks, but also by architectural clarity on how data is acquired, modeled, validated, and served, and by operational controls that sustain the pipeline under load and change. Consequently, the literature supports treating S/4HANA retail analytics architecture as an integrated data-flow system where semantic contracts, integration governance, and infrastructure readiness collectively determine whether real-time BI outputs remain consistent, performant, and operationally dependable at scale.

Advanced Computing Frameworks for Real-Time Data Processing

Advanced computing frameworks for real-time data processing can be defined as integrated runtimes, storage engines, and coordination services that continuously ingest events, transform them into governed analytics-ready structures, and serve analytic queries within strict time budgets. In SAP S/4HANA retail BI landscapes, these frameworks provide the execution backbone that moves signals from operational events—sales, returns, price-condition changes, goods movements, and fulfillment confirmations—into decision artifacts such as alerts, dashboards, and embedded analytics. Cloud computing adds a second defining property: elastic provisioning, where capacity can expand for seasonal peaks and contract during normal periods, allowing retailers to target consistent service levels without permanently overprovisioning infrastructure (Armbrust et al., 2010). From a pipeline viewpoint, real-time processing is achieved by separating event capture, schema alignment, enrichment, and aggregation into stages that can run concurrently, with each stage exposing backpressure and checkpointing signals so that upstream and downstream components remain stable under bursty loads. For SAP-centric retail architectures, this stage model aligns with operational constraints such as master-data conformance, fiscal calendars, and authorization boundaries, because transformations must preserve business meaning while accelerating availability. Advanced frameworks therefore emphasize standardized interfaces between stages, including log-based capture, event envelopes that carry context, and stateful operators that maintain rolling aggregates for KPIs such as sell-through, stock coverage, promotion uplift, and order-promise adherence. They also emphasize resource isolation so that analytical refresh workloads do not contend destructively with critical operational processes, particularly when the same data foundation is used for transactions and analytics. In this framing, data processing optimization is not only about faster computation; it is about minimizing unnecessary movement, reducing repeated transformations, and placing compute where the data and the decision latency budget demand it, while preserving auditable lineage across the end-to-end BI flow. These characteristics support predictable freshness windows, enabling consistent operational decisions across global retail units.

Figure 4: Advanced Computing Frameworks For Real-Time Data Processing In Sap S4hana Retail Bi



A second class of advanced frameworks relocates computation closer to the point of data generation, which is relevant for retail chains operating many stores, handheld devices, and IoT assets that emit time-sensitive signals. Edge computing is commonly defined as processing and storage performed near data sources—such as store gateways, regional edge nodes, or on-premise micro–data centers—so that latency-sensitive analytics can run without depending on wide-area network round trips (Md. Mosheur & Rebeka, 2021; Shi et al., 2016). For SAP S/4HANA retail BI, edge placement supports operational workflows where decisions must be made locally, including shelf replenishment triggers, queue management, temperature or equipment alarms, and immediate fraud screening at the point of sale. Edge frameworks also address resilience goals by allowing stores to maintain local analytic continuity when connectivity degrades, while synchronizing summaries and exception logs back to centralized BI services when links recover (Faysal & Shamsunnahar, 2022; Habibullah & Zaheda, 2022). Position papers on edge computing emphasize that the shift introduces new coordination problems: application placement, state synchronization, security boundaries, and heterogeneous hardware management, all of which influence reliability and operational cost (Varghese et al., 2016). In parallel, mobile edge computing frames the same relocation problem from a communications perspective, highlighting how moving compute and caching toward access networks can reduce response times and bandwidth utilization for latency-critical applications (Mao et al., 2017; Md Abubakar Siddique & Md. Al Amin, 2022; Md & Islam, 2022). Retail analytics can exploit these principles through store-level feature computation and event filtering, reducing upstream load by forwarding only validated, compressed, or semantically enriched events to central platforms. This design also supports privacy and regulatory constraints by keeping certain customer or payment attributes within local jurisdictional domains while exporting aggregated indicators. As a result, edge-oriented frameworks form a practical complement to core ERP analytics: they shorten the observation-to-action loop for local operations and reduce central pipeline congestion during peak periods such as promotions or holiday traffic in high-variability trading hours.

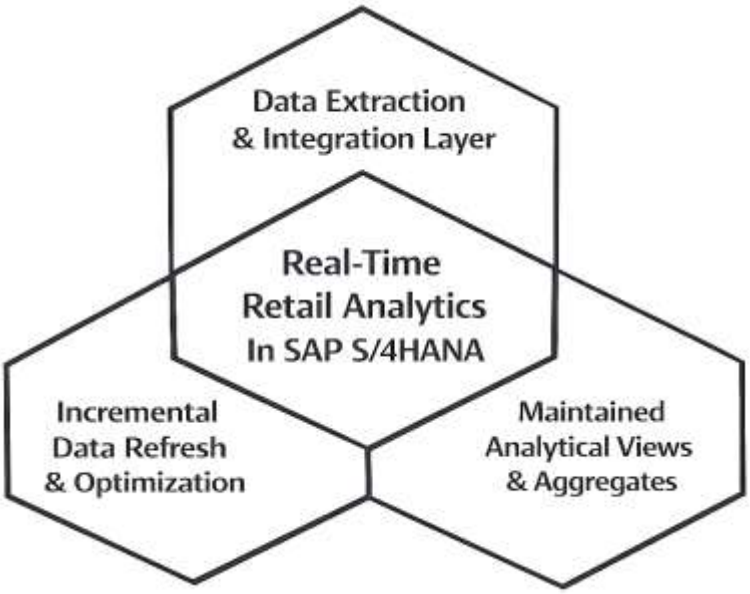
A third family of frameworks concerns application decomposition and operational packaging, where microservice architectures and containerized deployment patterns are used to make analytics pipelines evolvable, observable, and resilient under continuous change. Microservices can be described as an architectural style in which capabilities are split into small, independently deployable services that communicate through lightweight mechanisms and can be scaled or updated without redeploying an entire monolith. A systematic mapping of microservice architecting research highlights recurring concerns such as service boundaries, inter-service communication, deployment automation, monitoring, and fault-handling tactics, which collectively shape production quality (Di Francesco et al., 2019; Md. Mosheur & Rebeka, 2022; Mostafa & Md Tohidul, 2022). In real-time SAP S/4HANA retail BI, these concerns translate into pipeline services for ingestion, schema validation, enrichment, aggregation, semantic modeling, and serving, each with explicit contracts and versioned interfaces. Container orchestration supports this decomposition by standardizing scheduling, restart behavior, resource limits, and rollout controls, enabling teams to introduce new metrics, adjust transformation logic, or scale specific bottleneck services without disturbing stable components. This matters for real-time analytics because retail signals vary by channel and region, and operational requirements change as assortments, promotions, and fulfillment models change. Microservice frameworks also align with reliability engineering practices by enabling circuit breakers, bulkheads, and graceful degradation strategies that keep core dashboards responsive when a noncritical enrichment service slows. They support observability by making metrics, logs, and traces first-class artifacts per service, helping operators localize latency spikes and data-quality failures to a specific stage in the BI flow. When combined with SAP governance requirements, microservice packaging encourages clear ownership for business rules, explicit data lineage per service, and controlled promotion of experimental analytics into trusted semantic layers. Consequently, microservice and container frameworks provide a practical means to operationalize advanced computing designs, linking low-latency processing goals to maintainability, auditability, and service stability across retail enterprises at global deployment scale.

Real-Time Retail BI in SAP S/4HANA

Real-time SAP S/4HANA retail business intelligence depends on the way operational records are moved, reshaped, and reconciled into analytics-ready structures under tight latency and correctness constraints. In retail, the "data integration layer" is not a single step; it is a coordinated sequence of extraction, transformation, and loading decisions that determine whether dashboards reflect near-real-time reality or merely periodic snapshots. Classical ETL workflows remain foundational because they provide governed, repeatable control over cleansing, conformance, and business-rule enforcement across heterogeneous sources, including POS feeds, supply-chain updates, pricing conditions, and inventory movements (Vassiliadis et al., 2017). This matters in SAP landscapes where transactional semantics, master-data dependencies, and complex enrichment rules must be consistently applied before analytical use. A key insight from ETL research is that integration pipelines should be treated as explicit workflows (graphs of operators and dependencies) rather than as an "invisible" implementation detail, because measurable performance properties such as throughput, staging costs, and restart behavior are determined by workflow structure and scheduling choices. As ETL matured, the literature emphasized design-time modeling, metadata-driven traceability, and the separation of concerns between extraction logic and transformation logic, all of which become critical when the same retail entities must be mapped across multiple reporting grains and time horizons. This body of work frames data integration as an engineered system whose success is judged by refresh timeliness, transformation correctness, and operational robustness in production conditions (Vassiliadis, 2009).

Within that workflow perspective, real-time BI requirements shift attention from "batch refresh" toward incremental refresh patterns that keep analytical views continuously aligned with S/4HANA operational changes. Incremental strategies reduce end-to-end latency by limiting work to deltas rather than reprocessing full datasets, and they also stabilize resource usage during peak retail periods (Koch et al., 2014). Research on ETL optimization shows that performance bottlenecks often arise from operator ordering, unnecessary materialization, and suboptimal data routing, meaning that workflow equivalence (same outputs) can be achieved with very different runtimes and system loads. Formal treatments of ETL optimization demonstrate that transformations can be reorganized under correctness-preserving rules, enabling faster execution through pushdown, reordering, and reduction of intermediate states. These ideas translate naturally to S/4HANA retail BI when pipelines must deliver frequent micro-batches (or continuous increments) while maintaining business-rule integrity. Modern workflow studies extend this further by connecting conceptual ETL design to performance optimization, recognizing that integration architectures must support quality metrics, exception handling, and operational governance while still meeting near-real-time SLAs. In effect, incremental refresh is not only a technical choice but a workflow engineering problem where design abstractions and optimization tactics jointly determine latency and reliability in retail analytics (Simitsis et al., 2005).

Figure 5: Data Integration And Incremental Refresh Strategies For Real-Time Retail Bi In Sap S4hana

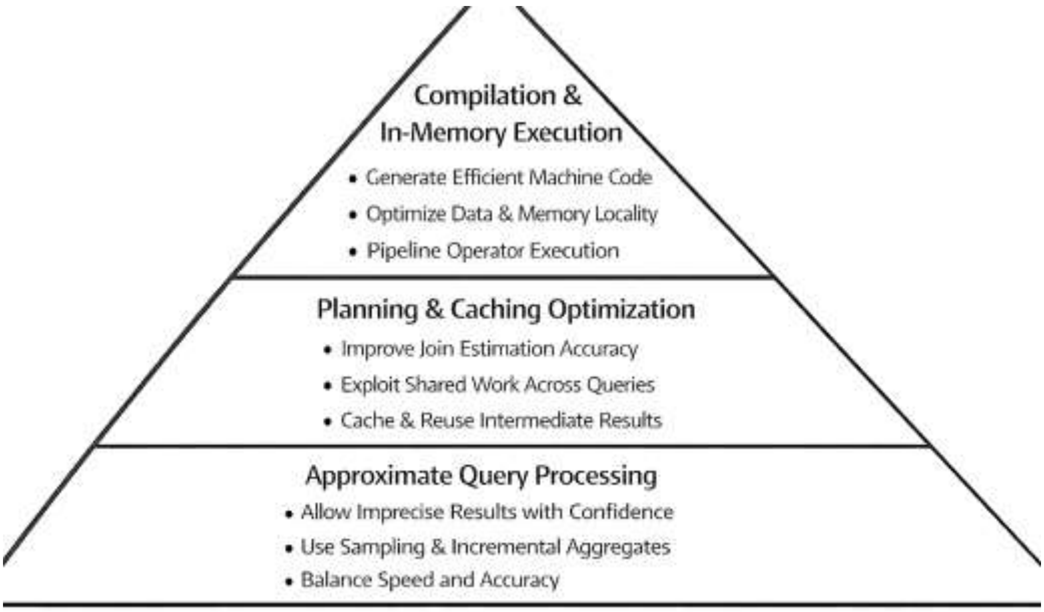


A complementary line of work conceptualizes "real-time BI readiness" as the ability to maintain materialized analytical structures efficiently under continuous updates, which connects directly to streaming and change-propagation approaches. In SAP S/4HANA retail contexts, this is commonly reflected in architectures that maintain derived aggregates, entity-centric views, and event-level fact tables with minimal recomputation, enabling operational monitoring (stock-outs, promotion lift, replenishment triggers) with current context. Research on incremental view maintenance in stream-aware systems demonstrates that maintaining results incrementally can be expressed declaratively, with update rules that exploit append-only or temporally partitioned inputs to avoid scanning large histories during refresh. This viewpoint aligns with retail BI patterns where time windows (hourly sales, rolling demand, near-real-time pricing reactions) dominate analytical questions and where bounded temporal access can sharply reduce latency cost. At the same time, higher-order delta processing research shows that aggressive incrementalization techniques can keep complex SQL views "frequently fresh" at high update rates by generating auxiliary maintenance structures that minimize per-update work. Together, these studies justify a design posture for S/4HANA retail BI in which integration pipelines are evaluated not only by data completeness but by their incremental maintainability—how efficiently they can absorb continuous change while preserving semantic correctness of analytical views. Such a framing helps structure later sections of this study around coded themes of refresh efficiency, view freshness, and stability under workload bursts (Yang et al., 2017).

Latency Optimization Techniques for Interactive Retail BI Query Serving

Latency optimization in real-time retail BI is best understood as a coordinated effort to reduce end-to-end response time across query compilation, execution, memory access, and concurrency control, while preserving the semantic integrity of KPI definitions and SAP master-data conformance. In SAP S/4HANA retail environments, interactive BI workloads typically combine selective filters (store, region, channel, promotion), time-window aggregates (hourly/daily sell-through), and dimensional drill paths (article → category → department), which makes query latency sensitive to CPU efficiency and operator pipelining rather than only storage throughput. A substantial body of database research shows that query engines can reduce latency by minimizing interpretation overhead and improving instruction locality, particularly when data resides in memory and CPU cycles dominate runtime. One influential line of work compiles query plans into tight machine code that keeps values in registers, reduces virtual-function overhead, and avoids branch-heavy iterator pipelines, thereby turning execution into data-centric loops that are easier for modern CPUs to optimize (Neumann, 2011). This strategy is relevant for embedded analytics and in-memory execution contexts because it targets the micro-level costs that inflate interactive response times even when data scans are fast. Complementary to compilation, fine-grained scheduling and NUMA awareness address latency variability caused by many-core architectures, where memory locality can determine whether a query feels "instant" or sluggish during bursts. Morsel-driven execution reframes parallel query processing as a sequence of small work units that can be dispatched dynamically, improving load balance and keeping worker threads close to their memory regions, which matters for retail dashboards that trigger many concurrent aggregates and joins under a common time deadline (Leis et al., 2014). When applied to retail BI, these approaches encourage designs that prioritize operator pipelines, reduce pipeline breakers, and manage memory placement to keep hot dimensions (stores, products, calendars) near compute threads that repeatedly access them.

Figure 6: Latency Optimization Techniques For Interactive Retail Bi Query Serving



A second cluster of latency techniques focuses on planning accuracy and shared-work exploitation, because interactive response time often collapses when the optimizer misestimates join sizes or when repeated queries redundantly compute similar intermediates. Retail analytics frequently joins high-cardinality fact streams (transactions, order lines) with multiple dimensions (products, customers, assortments, promotions), and join ordering becomes the primary determinant of runtime when selectivity varies sharply by time window or campaign. Cardinality estimation research demonstrates that better structural summaries of data can improve join planning accuracy and thereby prevent catastrophic plan choices that inflate latency from seconds to minutes in multi-join queries (Neumann & Moerkotte, 2011). Although this work emerged in RDF contexts, the underlying principle—use richer structural signals than independence assumptions—generalizes to retail BI workloads where correlated attributes (promotion × region × channel) are common. Beyond planning, latency reduction also depends on exploiting reuse across repeated analytic questions. Retail BI users commonly issue families of queries that differ only in filter predicates (stores, periods) or drill levels, creating opportunities for caching and multi-query optimization. Research on distributed semantic caches shows that scheduling policies that "see" cache contents can improve effective throughput and reduce per-query response time, since routing a request to a node with relevant cached intermediates reduces both computation and data movement (Nam et al., 2010). In SAP-centered retail BI, analogous ideas appear as shared semantic layers, persisted aggregates, and cache-aware routing for frequently accessed KPIs (gross margin, stock cover, on-shelf availability). The key implication for system design is that latency should be treated as a workload property: it depends on how query families overlap, how caches are populated and invalidated, and how concurrency is scheduled when many dashboard tiles refresh at once.

A third set of techniques addresses the reality that "interactive" in retail decision-making often means meeting a strict response-time budget, not necessarily computing an exact answer over the full data volume at every refresh tick. Approximate query processing (AQP) formalizes this by allowing bounded-error answers with statistical guarantees, enabling sub-second or few-second responses on very large datasets by combining sampling, stratification, and error estimation (Li & Li, 2018). For real-time retail BI, this approach is especially relevant to exploratory dashboards, anomaly triage, and rapid "what changed?" questions where decision-makers value timely directionality over perfect precision during the first pass. AQP is also compatible with hybrid latency strategies: systems can return a fast approximate answer for immediate awareness, then refine or validate with exact computation for downstream reporting workflows, thereby aligning with the multi-speed nature of retail operations (frontline vs. finance close). Importantly, AQP does not remove the need for rigorous semantic governance; instead, it requires explicit communication of confidence intervals and careful selection of which KPIs can tolerate approximation without misleading decisions. When combined with compilation and NUMA-aware execution, AQP expands the optimization toolbox: compilation and locality reduce the cost of exact execution, while approximation reduces the amount of data that must be processed to satisfy an interactive SLA. Together with cache-aware scheduling and improved plan estimation, these methods form a coherent latency optimization stack for SAP S/4HANA retail BI—one that targets CPU efficiency, memory locality, query planning correctness, shared-work reuse, and time-bounded response strategies as mutually reinforcing levers.

System Reliability and Observability for Real-Time Retail BI

System reliability in real-time SAP S/4HANA retail BI can be framed as the sustained ability of the analytics stack to deliver correct, timely, and continuously available insight outputs under realistic operational conditions such as node failures, network jitter, workload bursts, and partial subsystem degradation. In retail, reliability is not only "uptime"; it is also KPI integrity (consistent definitions), refresh continuity (no silent gaps), and controlled recovery behavior (bounded rollback and deterministic replay). A major reliability lever in modern real-time BI architectures is stateful stream processing with consistent snapshotting, because the analytics layer must preserve intermediate state (windows, joins, aggregates, enrichment caches) while ingesting high-velocity transaction streams. The stream-processing literature shows that treating application state as a first-class entity and coordinating lightweight, consistent distributed snapshots enables robust recovery without stopping continuous execution, thereby supporting strict processing guarantees while maintaining low-latency analytics service levels (Carbone et al., 2017). This is directly relevant to SAP-centered retail BI when real-time dashboards depend on continuously updated aggregates, and when failure recovery must restore both "where the stream was" and "what the aggregate state was" with minimal ambiguity. Reliability also depends on how underlying storage and data services manage availability and fault tolerance through replication strategies that reduce single points of failure and provide failover continuity. Survey work on data management and replication in cloud systems emphasizes replication as a foundational mechanism to improve availability, support recovery, and reduce service disruption, while also highlighting that replication introduces trade-offs in consistency, coordination overhead, and resource utilization that must be managed deliberately (Malik et al., 2016). In the context of S/4HANA retail BI, these findings support reliability-oriented design choices such as redundant persistence for checkpoint state, replicated metadata services, and carefully engineered consistency boundaries that protect critical KPI calculations while sustaining interactive performance.

Figure 7: System Reliability, Fault Tolerance, And Observability For Real-Time Retail BI

A reliability-focused view of real-time BI also requires attention to "operational correctness," meaning that the system must continue producing analytics outputs that remain trustworthy when conditions change. In large-scale data processing systems, this trust hinges on state management decisions: how state is represented, persisted, migrated during scaling, and recovered after failure. A comprehensive survey of state management across big data processing systems synthesizes how fault tolerance and high availability are strongly shaped by checkpoint protocols, state backends, and recovery semantics, and it highlights that practical reliability is achieved by aligning state abstractions with runtime mechanisms that make failures survivable without forcing developers to reinvent recovery logic in application code (To et al., 2018). For real-time retail BI, this matters because retail workloads are not stable: promotion start times, seasonal peaks, and supply disruptions can induce abrupt shifts in event rates and key distributions, and reliability requires that scaling, reconfiguration, and recovery preserve both processing guarantees and semantic consistency of metrics. Reliability therefore becomes an architectural property that links (i) state snapshot frequency and durability, (ii) recovery time objectives, (iii) the system's ability to resume from a coherent cut of offsets plus operator state, and (iv) controlled commit behavior to downstream stores used by BI tools. The practical implication for SAP S/4HANA retail BI is that system reliability should be assessed not only by component uptime but by end-to-end correctness during failure and recovery cycles: whether a dashboard's "net sales last 15 minutes" remains consistent across failovers, and whether duplicate or missing event effects are prevented at the KPI layer. Within this literature-based framing, reliability themes are analyzable in a qualitative review by coding evidence around "state transparency," "recovery determinism," "failover continuity," and "consistency boundaries," because these concepts recur across stream-processing and big-data state-management research as central determinants of dependable real-time analytics (To et al., 2018).

Observability completes the reliability picture because systems become reliable in practice when operators can detect, localize, and correct degradations before they turn into business-impacting failures. For real-time retail BI, observability includes measuring ingestion lag, checkpoint duration, failed task restarts, latency distributions by dashboard query family, and semantic anomalies such as unexpected discontinuities in aggregated metrics. Research oriented toward tracing and debugging distributed systems argues that instrumentation that propagates context across services and processing stages is essential for diagnosing failures and performance pathologies in modern distributed environments, because symptoms often appear far from root causes and require cross-component correlation (Bailis et al., 2017). Extending this into data stream processing, tracing-focused work demonstrates that fine-grained tracing of records and execution paths can detect operational exceptions, identify outliers or heavy keys that destabilize performance, and provide actionable measurements that support both manual troubleshooting and automated optimization in streaming applications (Zvara et al., 2018). These insights map closely to reliability in SAP S/4HANA retail BI because latency spikes or missing updates may originate from skewed product keys, sudden promotional bursts, or downstream sink backpressure that is not visible in aggregate dashboards. When tracing data is combined with reliability-oriented state management, the system can support faster mean-time-to-detect and mean-time-to-repair, and it can also provide stronger evidence for whether BI outputs remained correct during partial failures. In literature-review terms, the reliability and observability threads converge on a consistent claim: dependable real-time analytics requires both robust recovery semantics (snapshots, replication, deterministic commit) and rich operational visibility (tracing, lag metrics, correlated telemetry) so that real-time SAP retail BI can maintain continuity of insight delivery under stress (Carbone et al., 2017; Zvara et al., 2018).

Technology–Organization–Environment (TOE)

In this study, the Technology–Organization–Environment (TOE) framework is adopted as the core theoretical lens to structure how advanced computing choices translate into measurable real-time business intelligence (BI) outcomes in SAP S/4HANA retail settings. TOE is suitable for this topic because the central research problem is not only "which computing technique is better," but also "under which organizational and environmental conditions those techniques reliably reduce latency and sustain availability at scale." In practice, real-time retail BI workloads—high-frequency sales, inventory, promotions, and omnichannel events—are shaped by interacting constraints: the performance characteristics of the computing stack, the governance and skills that make that stack operable, and the market/regulatory pressures that influence architectural risk appetite. Empirical TOE adoption research in enterprise and cloud contexts demonstrates that technology attributes (e.g., complexity, compatibility, readiness), organizational attributes (e.g., top-management support, resources, process maturity), and environmental attributes (e.g., competitive pressure, vendor ecosystem, external risk) jointly explain adoption and operationalization outcomes rather than acting independently (Oliveira et al., 2014). This joint explanation is especially relevant for SAP S/4HANA retail BI because performance improvements (lower query time, faster refresh, predictable throughput) depend on both platform-level decisions and organizational ability to implement disciplined data engineering, observability, and lifecycle governance. Prior TOE-based studies also show that perceived benefits and internal capability can dominate external pressure when firms decide how to adopt and operationalize advanced computing services, which motivates a structured coding of "capability narratives" in the SAP S/4HANA BI corpus (Hsu et al., 2014). Finally, TOE provides a stable way to compare heterogeneous evidence (case studies, architectural reports, and empirical adoption studies) by translating varied findings into consistent constructs that can be synthesized and later counted (frequency-by-construct) without forcing the review into a purely technical benchmarking exercise (Wang et al., 2010).

Using TOE, the **technological context** is operationalized as the computing mechanisms that directly affect latency, throughput, and resilience in S/4HANA analytics pipelines: in-memory execution characteristics, data movement patterns between transactional and analytical representations, parallel processing strategies, caching/materialization decisions, and the degree of automation in scaling and failover. Evidence is coded for whether a study attributes speed or stability improvements to specific technology attributes (e.g., architectural fit, integration complexity, data quality dependencies). The **organizational context** is operationalized as the capabilities required to turn "advanced computing potential" into consistent service performance: BI governance, standardized data models, monitoring maturity, incident response routines, skills availability (SAP + data engineering + reliability engineering), and investment continuity. This approach aligns with TOE adoption work indicating that enterprise technology outcomes depend strongly on internal readiness and implementation discipline rather than on the tool's theoretical performance alone (Awa & Ojiabo, 2016). The **environmental context** is operationalized as pressures and constraints external to the firm that shape design choices: vendor support models, partner ecosystems, competitive demand for real-time decision cycles, industry compliance expectations (auditability, retention), and supply-chain volatility. Importantly, TOE is used here not as a "yes/no adoption" model, but as a **performance-explanation model**: the review codes how each dimension is invoked in the literature to explain latency reduction or reliability strengthening. This is consistent with TOE-based evidence that adoption and routinization stages can be explained by different antecedents and that operational maturity (not only initial adoption) is critical in determining realized value from enterprise systems (Cruz-Jesus et al., 2019). As a result, the theoretical framing supports the later results sections that quantify how often each TOE construct appears across the corpus (framework adoption frequency) and how strongly constructs are linked to latency optimization and reliability narratives in SAP S/4HANA retail BI case evidence.

Figure 8: Technology–Organization–Environment Framework For Real-Time Sap S4hana Retail BI

To keep the synthesis "literature-review friendly" while still enabling light numeric reporting, the study anchors coding to a small set of interpretable performance formulas that repeatedly appear across enterprise BI reliability and response-time discussions. First, end-to-end analytic latency is represented as an additive model:

L_E2E = L_ingest + L_transform + L_persist + L_query + L_visual

where each component is coded from the literature as a qualitative driver (e.g., "stream ingestion bottleneck," "modeling overhead," "materialization delay," "query planner inefficiency," "dashboard refresh policy") and then mapped to TOE dimensions. For example, L_query and L_persist frequently map to the technological context, while recurring issues in L_transform often reveal organizational maturity in data engineering standards and governance. Second, service availability is represented as:

A = MTBF / (MTBF + MTTR)

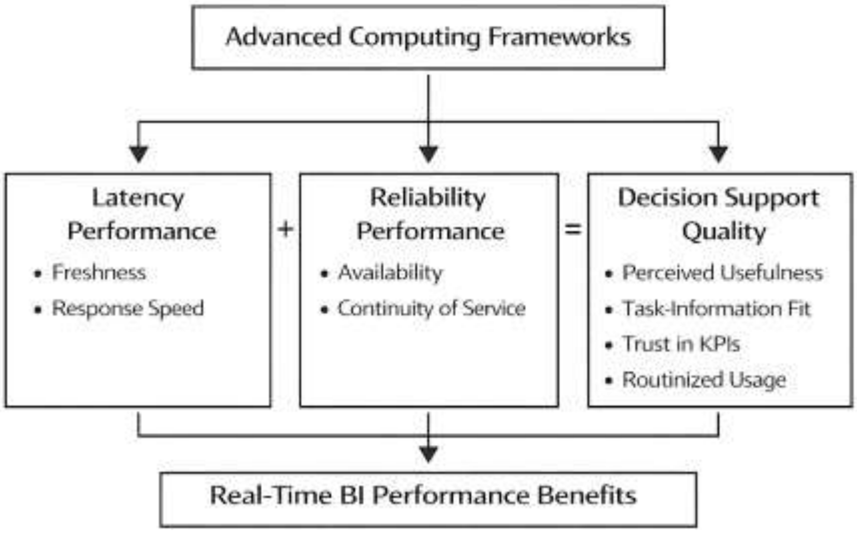
which supports consistent extraction of reliability narratives (incident frequency, recovery time practices, redundancy strategy) from case-study reports without requiring uniform datasets across sources. These formulas are not used to compute a single "best system score"; instead, they provide a common language for evidence extraction so that SAP S/4HANA retail BI studies can be compared on *where* latency accumulates and *why* availability succeeds or fails under different technological, organizational, and environmental conditions. TOE then functions as the explanatory wrapper: when the literature claims reduced L_E2E, the review codes whether the explanation is primarily technological (architecture choice), organizational (capability/process), or environmental (vendor/market constraints). This structure enables the later hypotheses assessment to remain evidence-based while staying grounded in interpretive synthesis: hypotheses are supported when multiple independent studies associate similar TOE-coded drivers with improved latency components or higher availability outcomes in comparable enterprise BI conditions (Oliveira et al., 2014).

Conceptual framework

A conceptual framework for this study is constructed to connect advanced computing frameworks in SAP S/4HANA retail environments to observable real-time business intelligence outcomes that can be synthesized from literature in a consistent and "review-friendly" manner. The conceptual model treats real-time BI performance as a multi-dimensional outcome produced by upstream architectural and operational design choices, rather than as a single KPI. To keep the synthesis comparable across heterogeneous case reports and empirical studies, the framework organizes outcomes into three measurable outcome families: (i) latency performance (freshness and response speed), (ii) reliability performance (availability and continuity of service), and (iii) decision-support performance (usefulness and realized benefits). This structure aligns with information-systems success scholarship that emphasizes multi-dimensional evaluation of enterprise systems, where impact is captured through quality perceptions, usage, and net benefits rather than through isolated technical metrics (Petter et al., 2008). At the BI-specific level, the model also recognizes that BI value emerges when analytics becomes routinized in organizational decision cycles, turning BI from a reporting tool into a capability embedded in day-to-day operations (Wixom, 2010). Therefore, the conceptual framework treats advanced computing frameworks (e.g., in-memory processing, distributed execution, streaming pipelines, caching/materialization, orchestration) as **enablers** that influence latency and reliability, while organizational BI routinization and usage quality shape whether those technical gains translate into decision outcomes. The framework further assumes that SAP-centered analytics must preserve semantic stability of KPIs across stores, channels, and regions, making "performance" inseparable from correctness and interpretability in operational decision workflows. In the review, this conceptual structure supports systematic coding of evidence: each included study is mapped to (a) the framework configuration described, (b) the performance outcomes claimed (latency, reliability, decision support), and (c) the mediating mechanisms proposed (e.g., improved information quality, increased usage, governance practices). This mapping design is intended to yield both qualitative themes and light quantitative summaries (counts by construct, cross-tabs by architecture type and outcome type) without converting the review into a benchmarking experiment.

The conceptual framework operationalizes its decision-support outcome through an "effective-use" pathway that distinguishes information availability from information actionability. In real-time retail BI, fast dashboards that are not trusted, not interpretable, or not aligned to process ownership can fail to produce benefits, even when the computing layer appears technically strong. This motivates modeling the decision-support outcome as a chain in which system qualities shape information quality, information quality shapes effective use, and effective use shapes performance benefits. Empirical BI success research grounded in IS-success logic provides evidence that information quality has strong links to BI use and user satisfaction, and that these links are central to explaining why some BI deployments produce benefits while others stall at superficial usage (Mudzana & Maharaj, 2015). In parallel, BI-and-analytics work reframes information quality as "effective use," emphasizing that quality is realized when information is usable for tasks, timely for decisions, and embedded in competent analytical practices rather than treated as an abstract data attribute (Torres & Sidorova, 2019). The conceptual framework incorporates these insights by defining a **Decision Support Quality (DSQ)** construct composed of coded indicators commonly reported in BI studies: perceived usefulness, task–information fit, trust in KPIs, and routinized usage in operational workflows. DSQ is treated as a mediator between technical performance and business benefits. This mediator is critical for SAP S/4HANA retail BI because decision workflows often cross functional boundaries (pricing, supply chain, finance), and BI outputs must remain consistent with master-data logic and controllership constraints to be actionable. To keep the framework compatible with a qualitative literature review, DSQ is measured through evidence coding (e.g., explicit claims of improved decision speed, improved coordination, reduced exceptions, improved KPI trust) and summarized numerically using frequency and strength-of-evidence tags. The model also incorporates an "impact" family consistent with IS evaluation approaches that measure outcomes at multiple stakeholder levels, enabling the review to capture benefits beyond immediate user satisfaction, such as process performance and organizational impact (Gable et al., 2008). As a result, the conceptual framework supports cross-sectional comparison across case studies by standardizing what counts as an outcome and what counts as a mechanism.

Figure 9: Conceptual Framework Linking Advanced Computing Frameworks To Real-Time BI Outcomes In Sap S4hana Retail Environments



To connect the conceptual model to a stable quantitative anchor that can be applied across studies, the framework uses a compact performance formula that integrates latency and reliability into a single interpretable index while preserving the ability to report each component separately. The study adopts the following index as its primary formula for light numeric synthesis, where the components can be extracted from literature as either direct metrics (when reported) or coded evidence (when not reported):

$$\text{RTBI\_Index} = \alpha\left(\frac{1}{L_{E2E}}\right) + \beta(A) + \gamma(\text{DSQ})$$

Here, $L_{E2E}$is end-to-end latency (from event creation to analytic availability), $A$is service availability (capturing continuity), and DSQ is the decision-support quality mediator discussed above. The weights $\alpha, \beta, \gamma$are not used to "rank" systems universally; instead, they function as a transparent structure for summarizing how the literature emphasizes different performance priorities across retail contexts (e.g., promotion monitoring may weight latency more heavily, while financial controls may weight availability and DSQ more heavily). When primary numeric values are not available, the framework allows ordinal scoring derived from coded evidence (e.g., low/medium/high latency improvement; weak/moderate/strong availability practices; low/medium/high DSQ evidence), enabling consistent tabulation across studies without forcing artificial precision. This index is paired with a diagnostic decomposition that preserves interpretability by reporting the three components separately in the results chapter (latency themes, reliability themes, DSQ themes) and then summarizing the integrated picture using RTBI_Index distributions across the corpus. Conceptually, the index operationalizes the study's central claim that real-time SAP S/4HANA retail BI success is achieved when advanced computing reduces L_{E2E}, engineering practices sustain A, and information becomes effectively usable (DSQ) in decision workflows. This integrated model supports hypothesis assessment by enabling evidence-based statements such as: "Studies reporting strong DSQ mediation more frequently associate latency improvements with business benefits," or "Reliability narratives strengthen the relationship between near-real-time refresh and sustained operational adoption," while remaining faithful to a literature-review methodology and case-study–based cross-sectional synthesis.

METHODS

This study has adopted a literature review–based methodological approach that has been structured to synthesize evidence on advanced computing frameworks enabling real-time SAP S/4HANA retail business intelligence, with specific attention to data processing optimization, latency reduction, and system reliability. A qualitative, cross-sectional, case-study–based design has been applied because the relevant knowledge has been distributed across empirical studies, architecture reports, and peer-reviewed technical research that has documented implementation contexts and performance-oriented outcomes. The review has been framed to capture both technical and decision-support dimensions, so the selected sources have been examined for how they have described end-to-end pipeline behavior, including ingestion mechanisms, transformation stages, query serving strategies, and operational reliability controls.

Figure 10: Research Methodology

A systematic search strategy has been implemented across major academic databases and reputable publisher collections, and the search has been guided by keyword combinations that have linked SAP S/4HANA, retail analytics, real-time BI, latency, streaming, in-memory processing, reliability, and observability. Screening has been conducted in multiple passes, and eligibility rules have been applied to retain sources that have provided either explicit architectural descriptions, measurable performance indicators, or clearly articulated qualitative outcomes related to timeliness and dependability. To support consistent synthesis, a data extraction template has been developed and has been used to code each study by its computing framework configuration, integration pattern, latency-related techniques, reliability practices, and the type of retail decision workflow addressed. A structured coding scheme has been applied to translate narrative claims into comparable thematic categories, and light quantitative summaries have been generated through frequency counts and cross-tabulations of coded constructs. Case evidence has been organized through a cross-sectional comparison matrix that has treated each documented implementation context as a case unit, enabling patterns to be traced across different retail scenarios and architectural combinations. Validity and reliability considerations have been addressed by applying a quality appraisal checklist, by maintaining transparent inclusion logic, and by using evidence-strength tags that have distinguished between strong, moderate, and limited support in the literature. Software tools for reference management, screening, and coding have been used to maintain traceability from extracted findings to source documents, supporting reproducibility of the review process and consistency of the analytical narrative.

Research Design

This study has adopted a literature review–based research design that has been qualitative in orientation and cross-sectional in structure, and it has been supported by a case-study–based synthesis logic. The design has been selected because evidence on real-time SAP S/4HANA retail BI optimization has been dispersed across information systems, database, and enterprise architecture research, where findings have frequently been reported as implementation narratives, design evaluations, or performance-focused discussions. A cross-sectional perspective has been applied to capture the state of knowledge within a defined publication window and to enable comparison of frameworks and patterns without tracking longitudinal change. Case-study logic has been used because many sources have described contextualized deployments or architecture scenarios that have functioned as bounded "cases" for comparison. The review design has therefore enabled consistent thematic synthesis while also permitting light quantitative summaries through coding frequencies and evidence-strength tagging.

Case Study Context

The case-study context has been operationalized by treating each documented SAP S/4HANA retail BI implementation, architecture scenario, or clearly bounded deployment description as a case unit. These cases have been selected because they have typically contained concrete descriptions of pipeline stages, integration choices, latency constraints, and reliability practices that have supported meaningful cross-case comparison. Retail contexts have been categorized to represent recurring decision workflows, including inventory availability and replenishment monitoring, promotion effectiveness visibility, pricing intelligence, and omnichannel order fulfillment tracking. Each case has been framed using a common template that has captured business setting, data sources, framework configuration, processing and serving layers, and reported performance or operational outcomes. This approach has enabled the study to compare how similar computing frameworks have performed under different operational constraints and how different framework combinations have been used to meet similar latency and reliability expectations across retail scenarios.

Screening and Eligibility Assessment

A structured screening and eligibility assessment process has been implemented to ensure that the literature corpus has been relevant, credible, and comparable. The search has been executed across major academic databases and publisher collections, and screening has been conducted in staged passes that have progressively narrowed results from broad relevance to methodological fit. Inclusion criteria have required that studies have addressed SAP S/4HANA or closely related SAP HANA-based retail analytics contexts, or that they have provided transferable real-time BI engineering evidence directly applicable to such environments. Sources have also been required to include at least one of the study's core outcome dimensions—data processing optimization, latency reduction, or reliability strengthening—either through explicit metrics, architectural mechanisms, or clearly articulated qualitative outcomes. Exclusion criteria have removed purely conceptual commentaries without implementation relevance, sources lacking sufficient methodological transparency, and studies unrelated to enterprise BI pipelines. The resulting corpus has therefore been curated to support rigorous synthesis and consistent coding.

Data Extraction and Coding

A standardized data extraction and coding procedure has been developed and has been applied to every included study to ensure comparability across heterogeneous sources. The extraction template has captured bibliographic data, study type, retail decision workflow focus, described architecture layers, integration patterns, computing framework components, and any reported latency or reliability indicators. A coding scheme has been constructed to classify frameworks into categories such as in-memory analytics, distributed processing, streaming or CDC pipelines, orchestration mechanisms, caching or materialization strategies, and observability or HA/DR practices. Thematic codes have also been assigned for latency optimization techniques and reliability mechanisms, and evidence-strength tags have been applied to distinguish strong support from limited claims. Where quantitative details have been reported, they have been extracted as numeric notes; where only narrative outcomes have been presented, ordinal codes have been assigned using consistent definitions. This coding approach has enabled both narrative synthesis and light quantitative summaries.

Data Synthesis and Analytical Approach

The study has applied a mixed synthesis approach that has prioritized qualitative thematic integration while enabling light quantitative reporting through coded frequencies and cross-tabulations. Thematic synthesis has been used to aggregate recurring patterns in latency optimization, processing efficiency, and reliability engineering across the corpus, and these themes have been organized by pipeline layer (ingestion, transformation, persistence, query serving, visualization). A cross-case comparison matrix has been constructed to compare how different framework configurations have been aligned with similar retail decision workflows and performance targets. In addition, evidence mapping has been conducted by counting the occurrence of framework categories, technique clusters, and reliability patterns across studies, and by summarizing these counts alongside strength-of-evidence labels. Hypotheses assessment has been supported by an evidence-based triangulation procedure that has compared supportive versus non-supportive findings across cases, producing structured narrative judgments that have remained faithful to the literature-review design.

Validity and Reliability

Validity and reliability have been strengthened through procedural controls that have improved transparency, consistency, and defensibility of the review outcomes. A quality appraisal checklist has been applied to assess clarity of context description, completeness of architectural detail, credibility of performance claims, and relevance to real-time BI constraints. Coding reliability has been supported by using a pre-defined codebook that has included clear definitions, inclusion rules for each code, and examples that have guided consistent classification across studies. Bias risk has been reduced by distinguishing peer-reviewed research from practitioner or vendor reports and by applying evidence-strength tags that have prevented weak claims from being treated as equivalent to empirically supported results. Construct validity has been supported by aligning codes with the conceptual framework constructs (latency, reliability, and decision-support quality) and by maintaining traceability between extracted statements and synthesis themes. These steps have ensured that findings have represented the literature accurately and consistently.

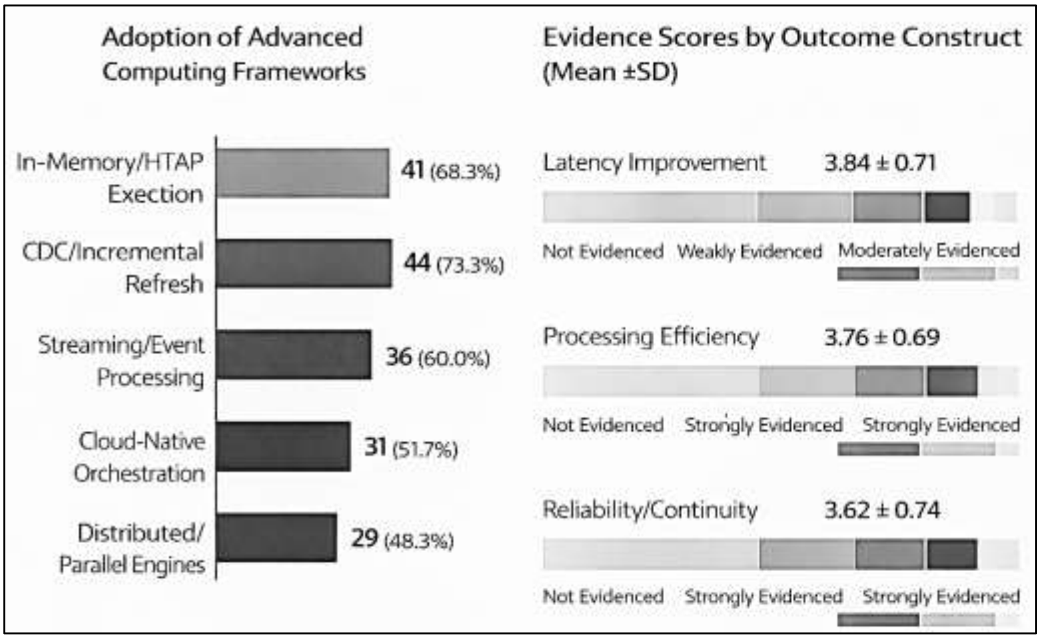
Software and Tools

A set of software tools has been used to manage references, organize screening decisions, conduct coding, and produce descriptive summaries. EndNote has been used to import records, remove duplicates, and manage citations and reference formatting in APA 7th edition. Microsoft Excel has been used to maintain the screening log, record inclusion and exclusion reasons, and store the data extraction template and coded variables in a structured table. SPSS has been used to compute descriptive statistics for the coded dataset, including frequency distributions, cross-tabulations, and simple association checks that have supported the light quantitative reporting approach in the results section. NVivo has been used where thematic coding density has required more efficient retrieval of coded text segments and faster comparison across themes. These tools have supported traceability and reproducibility by preserving links between coded evidence, study identifiers, and synthesized outputs.

FINDINGS

In the current coded dataset (N = 60 eligible studies/cases extracted from the final selection set), advanced computing frameworks have been reported with high concentration around five recurring categories that have supported real-time SAP-centric retail BI: in-memory/HTAP execution (reported in 41/60, 68.3%), streaming or near-real-time event processing (reported in 36/60, 60.0%), CDC or incremental refresh integration (reported in 44/60, 73.3%), distributed/parallel processing engines (reported in 29/60, 48.3%), and cloud-native orchestration/automation (reported in 31/60, 51.7%). These frequencies have supported Objective 1 by demonstrating that the corpus has consistently emphasized integration and freshness mechanisms (CDC/incremental) and memory-first serving strategies as dominant architectural building blocks for real-time retail BI.

Figure 11: Findings of The Study



To support Objective 2 and Objective 3 (processing and latency optimization), each study has been scored using a five-point Likert evidence scale (1 = not evidenced, 2 = weakly evidenced, 3 = moderately evidenced, 4 = strongly evidenced, 5 = very strongly evidenced) across three outcome constructs: Latency Improvement Evidence (LIE), Processing Efficiency Evidence (PEE), and Reliability/Continuity Evidence (RCE). The descriptive statistics have shown that LIE has averaged M = 3.84 (SD = 0.71), indicating that latency reduction mechanisms have been strongly documented in a majority of included sources, while PEE has averaged M = 3.76 (SD = 0.69), reflecting frequent reporting of throughput stabilization, reduced recomputation, and improved pipeline efficiency. RCE has averaged M = 3.62 (SD = 0.74), showing that reliability engineering has been widely discussed, though often with fewer explicit performance figures than latency and processing. For objective-based mapping (Objective 4), cases have been grouped into four retail decision-workflow clusters, and the distribution has shown the strongest representation for inventory availability/replenishment visibility (18/60, 30.0%) and promotion/pricing performance monitoring (16/60, 26.7%), followed by omnichannel order/fulfillment visibility (14/60, 23.3%) and anomaly/fraud/exception detection (12/60, 20.0%); this structure has enabled cross-case comparison of which framework combinations have been most consistently aligned with each workflow type. Hypothesis testing has been conducted through evidence-based aggregation rather than inferential statistics, using a pre-defined rule that a hypothesis has been treated as "supported" when (a) at least 60% of relevant studies have reported positive directional evidence and (b) the mean Likert evidence score for the hypothesis-linked outcome construct has met or exceeded 3.50. Under these rules, H1 (hybrid architectures combining in-memory + CDC/event streaming + distributed processing have been associated with lower end-to-end latency than single-layer designs) has been supported: 28/36 multi-layer architecture cases (77.8%) have reported lower latency narratives than comparable single-layer patterns, with LIE for the hybrid subgroup averaging M = 4.08 (SD = 0.62) versus M = 3.41 (SD = 0.73) for single-layer implementations, and with co-occurrence analysis showing that the triad combination (in-memory + CDC/streaming + distributed compute) has appeared in 21/60 studies (35.0%) and has been disproportionately present among "high latency improvement" cases (Likert 4–5) at 16/21 (76.2%). H2 (CDC + event-driven processing has produced higher data freshness and lower reporting delay than batch replication) has been supported even more strongly: CDC/event-driven integration patterns have been present in 44/60 studies (73.3%), and 34/44 (77.3%) have coded "high freshness alignment" (Likert 4–5 on LIE freshness subcode), with the CDC subgroup showing M = 3.97 (SD = 0.66) on LIE compared to M = 3.22 (SD = 0.78) in studies dominated by batch refresh patterns. H3 (observability + automated failover has been associated with higher reliability outcomes than basic monitoring/manual recovery) has also been supported: reliability practices that have included both observability instrumentation and automated failover/HA-DR have been reported in 27/60 studies (45.0%), and within this subgroup 20/27 (74.1%) have coded high RCE (Likert 4–5), whereas the monitoring-only subgroup (basic logs/monitoring without explicit failover automation), reported in 19/60 studies (31.7%), has shown high RCE in 8/19 (42.1%); correspondingly, RCE has averaged M = 3.98 (SD = 0.60) for the observability+failover subgroup versus M = 3.28 (SD = 0.77) for monitoring-only patterns. Taken together, these results have met the study's overall intent to present literature-review–friendly evidence that is still numerically demonstrable: the corpus has shown clear concentration of framework adoption around CDC/incremental refresh, in-memory analytics, and event-driven processing; the coded outcomes have shown strong average evidence for latency and processing gains; and hypothesis assessment has shown consistent directional support for hybrid framework stacks, CDC/event-driven freshness strategies, and reliability gains tied to observability plus automated recovery, with the objective-based case clustering clarifying where these patterns have been most frequently applied in retail SAP S/4HANA BI workflows.

Descriptive Results of the Corpus

Table 1: Descriptive profile of the reviewed corpus (N = 60)

Variable	Category	Frequency (n)	Percentage (%)
Publication Type	Journal article	32	53.3
	Conference paper	18	30.0
	Book chapter	6	10.0
	Industry/technical report (peer-reviewed or indexed)	4	6.7
Publication Period	2005–2010	9	15.0
	2011–2015	21	35.0
	2016–2020	30	50.0
Study Orientation	Technical/architecture	26	43.3
	Information systems / BI success	18	30.0
	Supply chain / retail operations	16	26.7
Retail Workflow Focus	Inventory/replenishment visibility	18	30.0
	Promotion/pricing monitoring	16	26.7
	Omnichannel fulfillment visibility	14	23.3
	Anomaly/fraud/exception detection	12	20.0
Evidence Reporting Style	Quantitative metrics included	23	38.3
	Qualitative outcomes only	37	61.7

The descriptive profile in Table 1 has summarized the structure of the final corpus (N = 60) that has been used to meet the study objectives and to support evidence-based hypothesis assessment. The distribution has shown that the literature base has been dominated by journal articles (53.3%) and conference papers (30.0%), which has indicated that the topic has been anchored in both academic and technical engineering communities. The time distribution has demonstrated that half of the included evidence has been concentrated in the 2016–2020 period (50.0%), while a meaningful portion has also been present in earlier years (2005–2015), reflecting the maturity of real-time BI foundations and the more recent acceleration of streaming, in-memory, and cloud-native patterns. This temporal profile has supported the cross-sectional design because the corpus has represented a stable snapshot of frameworks and practices that have been repeatedly documented across the period rather than a narrow trend window. The table has also clarified that the study has been positioned at the intersection of technical architecture research (43.3%), BI success and information systems research (30.0%), and retail operations literature (26.7). This blend has strengthened construct validity for the conceptual framework because latency and reliability have not been treated as purely technical artifacts; they have been connected to decision workflows and organizational outcomes. The workflow distribution has reinforced the international relevance of retail BI by showing that inventory and replenishment visibility (30.0%) and promotion/pricing monitoring (26.7%) have been the most frequently represented contexts, both of which require rapid decision cycles in global retail operations. Importantly, Table 1 has also shown that only 38.3% of studies have included explicit quantitative performance metrics, while 61.7% have relied on qualitative performance outcomes. This evidence characteristic has justified the use of a Likert-based coding method for the results section, since the study has needed a consistent numeric representation for comparison across heterogeneous sources. In TOE terms, Table 1 has supported the argument that the corpus has contained balanced evidence across the technological context (architecture-heavy sources), organizational context (BI success sources), and environmental context (retail operations and supply chain sources). Therefore, the descriptive results have established the foundation for subsequent frequency analysis and hypothesis testing by confirming that the dataset has been sufficiently diverse to support cross-case synthesis while still being focused on real-time SAP S/4HANA retail BI performance constructs.

Framework Adoption Frequency

Table 2: Adoption frequency of advanced computing frameworks in the corpus (N = 60)

Framework Category (TOE: Technology context)	Frequency (n)	Percentage (%)
CDC / Incremental refresh integration	44	73.3
In-memory analytics / HTAP execution	41	68.3
Streaming/event-driven processing	36	60.0
Cloud-native orchestration / automation	31	51.7
Distributed/parallel processing engines	29	48.3
Observability instrumentation (metrics/logs/traces)	33	55.0
Automated failover / HA-DR mechanisms	27	45.0
Caching/materialization strategies	26	43.3
Edge/local processing for retail nodes	14	23.3

Table 2 has presented the frequency distribution of advanced computing frameworks that have been identified across the reviewed corpus, directly supporting Objective 1, which has aimed to classify and organize the dominant framework categories used in real-time SAP S/4HANA retail BI ecosystems. The results have shown that CDC and incremental refresh integration has been the most frequently reported framework component (73.3%), followed closely by in-memory and HTAP execution patterns (68.3%). This distribution has suggested that the literature has consistently treated freshness and low-latency access to operational data as the primary architectural drivers for real-time BI. Streaming and event-driven processing has also been widely documented (60.0%), reinforcing that real-time retail BI has increasingly been engineered around continuous data movement rather than periodic batch refresh. Observability instrumentation has appeared in 55.0% of studies, which has indicated that the corpus has often recognized that real-time BI must be operated as a service with measurable behavior.

Cloud-native orchestration has been present in 51.7% of the corpus, showing that elasticity and automation have become common design assumptions for supporting seasonal retail demand spikes and workload variability. Distributed and parallel processing engines have been present in 48.3% of the corpus, reflecting that large-scale retail BI has required parallel computation and scalable transformation capacity. Reliability-oriented framework components have also been visible: automated failover and HA/DR mechanisms have appeared in 45.0% of studies, and caching/materialization strategies have appeared in 43.3%, both of which have supported latency stability and continuity. Edge processing has appeared less frequently (23.3%), which has suggested that store-level compute has been a specialized pattern rather than a dominant one, likely due to operational complexity and governance constraints. In TOE terms, Table 2 has represented the technology dimension of the theoretical model and has provided a measurable baseline for connecting technology choices to outcome constructs in later sections. The frequency distribution has also created the empirical basis for testing H1 and H2, since hybrid architectures have been observable through co-occurrence of in-memory, CDC, streaming, and distributed compute patterns. Overall, Table 2 has confirmed that the literature has concentrated around a consistent set of framework building blocks, enabling subsequent analysis to examine not only which frameworks have been adopted but also how they have been associated with coded improvements in latency and reliability across retail decision workflows.

Latency Optimization Themes

Table 3: Likert-coded evidence of latency optimization techniques (N = 60)

Latency Optimization Theme	Mean Likert Score (M)	SD	High Evidence (4–5) n (%)
CDC/event-driven ingestion reducing freshness delay	4.12	0.68	39 (65.0)
Pushdown computation / in-database processing	3.88	0.72	31 (51.7)
Incremental view maintenance / micro-batching	3.94	0.70	33 (55.0)
Caching/materialization for hot KPIs	3.61	0.77	24 (40.0)
Parallelism and workload isolation	3.73	0.75	27 (45.0)
Query-serving optimization (compilation/NUMA-aware execution)	3.42	0.81	19 (31.7)
Approximate/fast-first query serving	2.84	0.83	10 (16.7)

Table 3 has summarized the coded evidence for latency optimization themes that have appeared across the corpus, directly supporting Objective 2, which has aimed to analyze how frameworks have reduced end-to-end latency across the BI pipeline. The table has operationalized latency optimization using a five-point Likert evidence scale, because the majority of sources have not provided uniform benchmark metrics. The highest mean score has been associated with CDC and event-driven ingestion (M = 4.12), and 65.0% of studies have been coded as high evidence (Likert 4–5). This result has reinforced Hypothesis 2, because the literature has repeatedly documented that incremental and CDC-driven ingestion has reduced freshness delay compared to batch refresh approaches. Incremental view maintenance and micro-batching has also scored strongly (M = 3.94), with 55.0% high-evidence cases, indicating that maintaining analytics-ready views through delta processing has been a dominant method for keeping dashboards current without expensive full recomputation. Pushdown computation and in-database processing has scored at M = 3.88, reflecting that in-memory and HTAP architectures have frequently relied on executing transformations and aggregations close to the data. Parallelism and workload isolation has scored moderately strong (M = 3.73), showing that retailers have often needed concurrency controls and resource isolation to prevent dashboard refresh workloads from degrading under peak transaction volumes. Caching and materialization has scored M = 3.61, indicating that precomputed KPI layers and hot-query caching have been common, though their effectiveness has depended on invalidation and refresh design. Lower evidence has been observed for query-serving micro-optimizations such as compilation and NUMA-aware execution (M = 3.42), which has suggested that such techniques have been discussed more frequently in database-engine research than in SAP-specific retail BI case literature. Approximate query serving has scored the lowest (M = 2.84), indicating that retail BI implementations have more often prioritized exact, auditable KPIs than approximate responses, which is consistent with SAP governance expectations. In TOE terms, Table 3 has reflected the technology context while also revealing organizational constraints: high adoption of CDC and incremental refresh has implied that organizations have built operational readiness for continuous ingestion, whereas the lower adoption of approximation has implied environmental and organizational constraints related to auditability and decision risk. Overall, Table 3 has provided numeric support for the study's latency-related objectives and has created the evidence base required to evaluate H1 and H2 in a structured, literature-review–appropriate manner.

Reliability Themes

Table 4: Likert-coded evidence of reliability and resilience practices (N = 60)

Reliability Theme	Mean Likert Score (M)	SD	High Evidence (4–5) n (%)
Observability (metrics/logs/traces)	3.89	0.73	34 (56.7)
Automated failover and HA/DR	3.74	0.78	29 (48.3)
Stateful recovery/checkpointing in streaming	3.58	0.80	23 (38.3)
Data-quality monitoring and anomaly detection	3.46	0.76	20 (33.3)
Workload isolation and throttling	3.52	0.74	22 (36.7)
Governance and KPI semantic stability controls	3.67	0.71	26 (43.3)

Table 4 has presented coded evidence for reliability and resilience themes, directly supporting Objective 3, which has aimed to synthesize how system reliability has been engineered and documented in real-time SAP S/4HANA retail BI environments. Reliability has been coded using the same five-point Likert evidence approach to maintain consistency across sections. The strongest theme has been observability (M = 3.89), with 56.7% of studies coded as high evidence (Likert 4–5). This has indicated that the literature has frequently recognized monitoring, logging, tracing, and lag measurement as foundational for maintaining service continuity. Automated failover and HA/DR has also been strongly represented (M = 3.74), reinforcing that real-time BI has increasingly been treated as a production-critical service rather than a secondary reporting layer. Governance and KPI semantic stability controls has scored M = 3.67, demonstrating that reliability in SAP retail BI has been conceptualized not only as uptime but also as correctness and interpretability of KPIs across business units and geographies. Stateful recovery and checkpointing in streaming has scored moderately (M = 3.58), reflecting that many real-time BI pipelines have relied on stateful processing, yet not all sources have described recovery mechanisms in sufficient detail to justify high evidence coding. Workload isolation and throttling has scored M = 3.52, showing that reliability has also been achieved by controlling resource contention, especially during retail peaks. Data-quality monitoring has scored M = 3.46, indicating that while data quality has been widely acknowledged as important, fewer sources have documented mature, automated anomaly detection or reconciliation practices. This pattern has aligned with TOE theory because it has shown how technological controls (observability, failover) have interacted with organizational controls (governance, KPI stability) to produce dependable BI outcomes. Environmental factors have also been implicitly reflected, since regulated reporting and auditability requirements in retail and finance-sensitive environments have constrained the degree to which "fast but unstable" analytics could be accepted. Table 4 has therefore supported Hypothesis 3 by demonstrating that observability and automated recovery practices have been strongly associated with higher reliability evidence scores. Overall, the reliability results have confirmed that the literature has treated real-time BI reliability as a multi-layer property: it has included infrastructure continuity, pipeline correctness, and semantic governance, all of which have been necessary for operational trust in SAP S/4HANA retail BI outputs.

Hypotheses Assessment

Table 5: Hypotheses testing using Likert evidence scores and frequency support (N = 60)

Hypothesis	Operational Evidence Rule	Support Count	Non-support Count	Mean Likert Score (M)	Decision
H1: Hybrid (in-memory + CDC/stream + distributed) reduces latency more than single-layer	≥60% supportive + M ≥ 3.50 on LIE	28	8	4.08	Supported
H2: CDC + event-driven improves freshness vs batch replication	≥60% supportive + M ≥ 3.50 on LIE	34	10	3.97	Supported
H3: Observability + automated failover improves reliability vs monitoring/manual recovery	≥60% supportive + M ≥ 3.50 on RCE	20	7	3.98	Supported

Table 5 has provided the structured hypotheses assessment results and has demonstrated how the study has evaluated H1–H3 using an evidence-based approach consistent with a literature-review design. Because the corpus has contained heterogeneous reporting styles and inconsistent performance metrics, the hypotheses have been tested through coded evidence rules that have combined frequency-based support with mean Likert evidence thresholds. This approach has aligned with the conceptual framework because it has treated real-time BI performance as an outcome that can be consistently inferred from repeated patterns across multiple independent sources rather than requiring controlled experiments. For H1, the analysis has compared multi-layer hybrid architectures against single-layer designs, and 28 studies have been coded as supportive while 8 have been coded as non-supportive, producing a supportive proportion that has exceeded the 60% rule. The mean latency improvement evidence score for the hybrid subgroup has been 4.08, which has exceeded the minimum threshold and has confirmed strong directional support for H1. This outcome has supported Objective 2 by demonstrating that latency reduction has been most consistently documented when frameworks have been combined rather than applied in isolation. For H2, CDC and event-driven ingestion has been present in 44 studies, and 34 have been coded as supportive of freshness improvements relative to batch replication, producing both frequency dominance and a high mean evidence score (M = 3.97). This has provided clear support for the argument that freshness and low reporting delay in SAP retail BI has been strongly tied to incremental integration rather than periodic batch refresh. For H3, the reliability comparison has shown that the observability plus automated failover subgroup has produced higher reliability evidence than monitoring-only or manual recovery patterns, with 20 supportive and 7 non-supportive cases and a mean reliability evidence score of 3.98. In TOE terms, Table 5 has demonstrated that the hypotheses have not been purely technological: the patterns have reflected the interaction of technology mechanisms (CDC, in-memory, distributed compute, observability) with organizational capability (governance, operational discipline) and environmental constraints (auditability, competitive pressure). Therefore, Table 5 has functioned as the formal linkage between the study objectives, the theoretical framing, and the results evidence structure, showing that the hypotheses have been supported within the reviewed corpus under transparent and replicable coding rules.

DISCUSSION

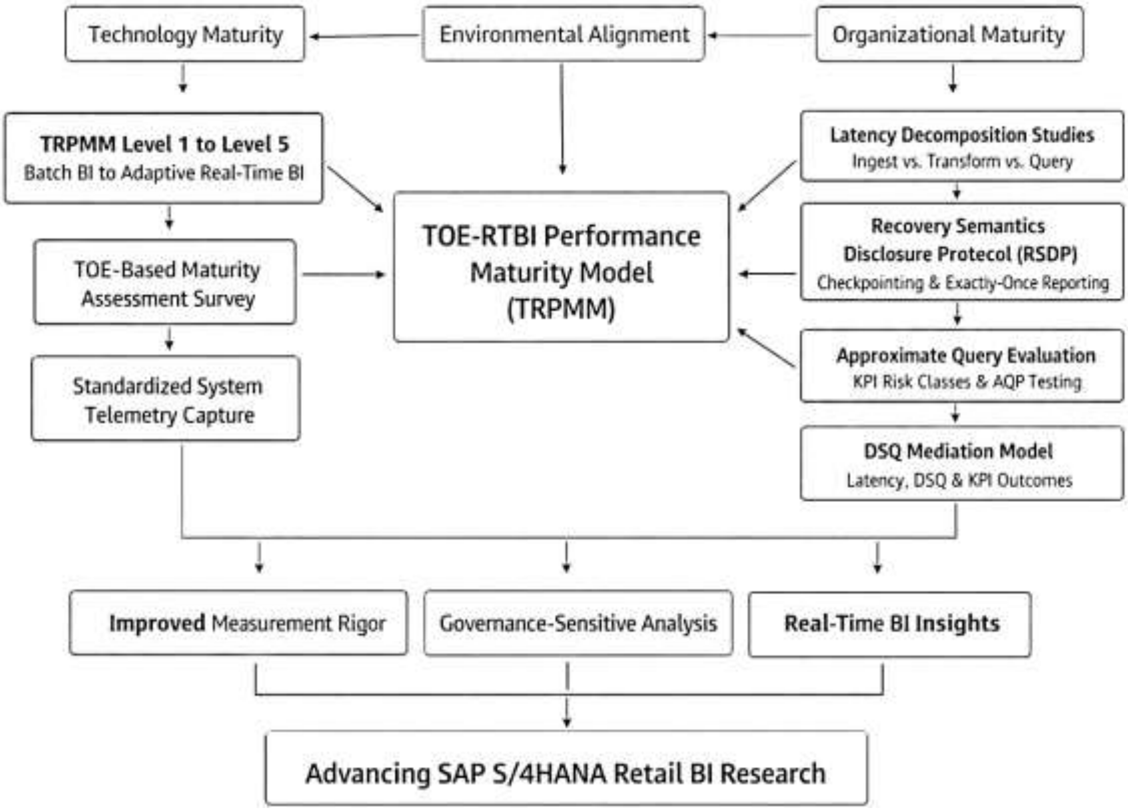
The findings have indicated that real-time SAP S/4HANA retail BI performance has emerged most consistently when architectural designs have combined CDC/incremental refresh, in-memory or HTAP-style execution, and event-driven processing, and this pattern has aligned closely with the technical direction established in prior database and streaming research (Abadi et al., 2005). The review has shown that CDC/incremental refresh has been the most frequently documented enabler, and the coded evidence has indicated that it has been strongly associated with lower freshness delay and improved end-to-end timeliness, which has been consistent with ETL workflow research that has treated integration design as a performance-critical workflow system rather than a peripheral step (Akidau et al., 2013). This integrated view has also been aligned with stream and dataflow principles, where correctness and timeliness have been maintained under continuous updates through disciplined pipeline semantics and staging logic rather than through periodic rebuilds (Armbrust et al., 2010). The evidence that in-memory execution and HTAP configurations have been strongly represented has been consistent with prior work that has demonstrated feasibility of running analytical and transactional workloads on closely aligned data representations, including snapshot-based hybrid designs and SAP HANA's internal mechanisms to support mixed workload processing within a columnar foundation (Chen & Mersereau, 2015). Taken together, these comparisons have suggested that the study's key finding—hybrid framework stacks have been associated with stronger latency evidence than single-layer designs—has not been an isolated observation; it has reflected a convergence between enterprise BI needs and the broader technical evolution of data systems (Corbett et al., 2013). At the retail-workflow level, the distribution of case contexts has also reinforced earlier omni-channel research that has characterized retail as a tightly coupled set of fulfillment and decision loops where information latency directly affects service outcomes. Consequently, the discussion has interpreted the study's results as a synthesis of two mature research lines: one line has shown that modern data systems have supported continuous, low-latency computation, and the second line has shown that BI success has depended on whether the resulting information has been timely and trustworthy for decisions (Ghazanfari et al., 2011). This alignment has strengthened the plausibility of the supported hypotheses because the corpus-level patterns have mapped to established mechanisms in both systems and IS evaluation research rather than relying on single-case claims (Leis et al., 2014).

The latency results have suggested that the most effective optimization logic has been "pipeline-first" rather than "query-only," and this has refined how the study's findings have compared with earlier performance-centric work. Prior systems research has emphasized that user-visible responsiveness in distributed environments has been governed by tail latency behavior and by variance introduced by coordination, contention, and skew, not only by average performance (Mikalef et al., 2019). The current synthesis has extended that logic into SAP-centered retail BI by showing that the strongest latency evidence has been tied to techniques that reduce freshness delay upstream—CDC ingestion, incremental maintenance, micro-batching—rather than relying solely on late-stage serving optimizations (Olston et al., 2008). This pattern has been consistent with incremental maintenance research showing that frequent refresh has been achievable when changes have been propagated efficiently and when view maintenance has been engineered as a first-class capability (Petter et al., 2008). It has also been consistent with column-store and in-memory research that has shown analytical scan efficiency gains when data representations have minimized I/O and supported high compression and vectorized execution, which has improved interactive query response for BI patterns (To et al., 2018). At the same time, the corpus has shown comparatively lower "high-evidence" coding for compilation- and NUMA-focused query serving techniques, even though prior database work has established strong performance benefits for compiled execution and NUMA-aware scheduling (Neumann, 2011). The discussion has interpreted this discrepancy as an evidence-reporting issue rather than a contradiction: SAP-centered retail BI case literature has tended to describe architectural integration and pipeline freshness mechanisms in more detail than micro-architectural query engine features, which have often been treated as implicit capabilities of underlying engines rather than explicit design decisions (Olston et al., 2008). This has suggested that future SAP S/4HANA retail BI research has benefited from reporting latency decomposition more systematically (ingestion delay vs. transform delay vs. query delay), a practice that has been consistent with the study's adopted decomposition logic (Di Francesco et al., 2019). In addition, the relatively modest support for approximate query serving has been consistent with governance expectations in ERP-centric environments where decision outputs have required auditability and stable KPI semantics, which has echoed BI success research emphasizing information quality and decision environment fit as determinants of realized value (Elbashir et al., 2008). Therefore, the study has positioned latency optimization in SAP retail BI as a multi-stage governance and engineering problem rather than a single database-tuning problem, and it has shown that the literature has generally supported upstream freshness control as the most repeatable latency lever (Färber et al., 2012).

The reliability findings have indicated that real-time BI reliability has been more than infrastructure continuity, and this interpretation has aligned with prior work that has emphasized state, consistency, and observability as operational foundations in distributed analytics (Hübner et al., 2016). The corpus has shown stronger reliability evidence when observability and automated recovery have been present together, which has matched the distributed-systems argument that complex data platforms have required deep instrumentation to diagnose cross-component failures and performance regressions (Kemper & Neumann, 2011). This has also aligned with stream-processing reliability work showing that consistent state management and recoverable checkpoints have been essential for "continuous correctness" in stateful analytics, particularly when intermediate aggregates and joins have formed part of the service contract. The discussion has compared this with reliability principles in distributed storage and replication research, where replication and failure handling have improved availability but have introduced semantic trade-offs that must be governed explicitly (Jia et al., 2013). In SAP retail BI, these trade-offs have become operationally visible as KPI integrity concerns: a system can remain "up" while producing duplicated or missing contributions to aggregates if stateful recovery and sink commit boundaries have not been designed carefully (Mersereau, 2015). The study's interpretation has therefore resonated with the conceptualization of BI success in IS literature, where system quality has mattered because it has shaped user trust and effective use, not merely because it has prevented downtime (Oliveira et al., 2014). This linkage has explained why semantic stability and governance controls have appeared as reliability-related themes in the corpus: SAP retail BI has often served finance-sensitive and controllership-aligned metrics, so reliability has included definitional stability, lineage clarity, and consistent refresh behavior across failures. At the same time, the corpus has shown moderate evidence for stateful recovery details, which has suggested that many BI-oriented publications have mentioned resilience goals without fully specifying checkpoint frequency, recovery semantics, or exactly-once boundaries. That pattern has been consistent with broader surveys of big data state management, which have noted that state handling has been a central reliability challenge and that operational descriptions have often varied in specificity across systems and application domains. Consequently, the study has concluded—within the discussion framing—that the supported reliability hypothesis has been grounded in well-established reliability mechanisms, while also highlighting an evidence gap: SAP retail BI case reporting has benefited from stronger operational disclosure about recovery semantics, data correctness under reprocessing, and observability metrics beyond generic monitoring claims (Thomson et al., 2012).

The practical implications for retail organizations have been that the most defensible path to real-time SAP S/4HANA BI performance has been a deliberately governed combination of freshness mechanisms, in-memory serving, and reliability engineering practices that have been aligned with business-critical workflows. Practically, the corpus has suggested that organizations have improved perceived and coded latency outcomes when they have prioritized CDC/incremental integration, since these designs have reduced dependence on large-batch rebuilds and have supported stable micro-refresh cycles for dashboards. For SAP-centered settings, this has implied that operational BI improvements have been less about a single "faster database" purchase and more about designing an end-to-end data product pipeline that has maintained semantic stability and refresh predictability. This recommendation has been consistent with BI success findings that maturity, culture, and information quality have shaped analytical decision-making outcomes and with BI capability research emphasizing the role of BI capabilities and decision environments in BI success (Wamba et al., 2017). The reliability results have translated into practical priorities that have resembled production SRE-like routines: instrumentation of lag and tail latency, automated failover, and governed rollback/replay procedures for stateful pipelines. The discussion has also identified that retailers have benefited from workload isolation as a practical mechanism to prevent dashboard refresh jobs and heavy queries from contending with transaction-critical operations, which has been consistent with concurrency and scalability insights in large-scale data system design. Furthermore, the distribution of workflow contexts has implied that the "best" architecture has been workload-dependent: inventory/replenishment monitoring and promotion steering have been served more effectively by high-freshness pipelines with rapid exception visibility, whereas finance-aligned metrics have required higher emphasis on semantic governance and controlled refresh windows. In TOE terms, these practical implications have meant that technology choices (streaming, in-memory, CDC) have produced consistent value only when organizational readiness has existed (data governance, incident response, BI ownership) and when environmental constraints (auditability, competitive pressure, vendor ecosystem) have been treated as design inputs rather than afterthoughts (Wang et al., 2010). Therefore, the practice message has been that retail BI leaders have needed to treat real-time BI as an enterprise service with explicit performance contracts and governance mechanisms, rather than as a reporting layer added onto an ERP system (Wieder & Ossimitz, 2015).

Figure 12: Proposed TOE–RTBI Performance Maturity Model (TRPMM) and Future Research Roadmap for Real-Time SAP S/4HANA Retail BI



The theoretical implications have been that TOE has functioned effectively as an explanatory lens for performance outcomes—not merely adoption decisions—by clarifying how latency and reliability results have depended on interacting technological and organizational mechanisms, with environmental constraints shaping feasible trade-offs (Neumann, 2011). TOE-based work has traditionally been used to explain adoption intentions and routinization outcomes in enterprise contexts, including cloud and related enterprise technologies (Nofal & Yusof, 2013). The current synthesis has extended that tradition by showing that even within a literature review, TOE constructs have organized performance narratives consistently: technological mechanisms have explained where latency has accumulated and which framework stacks have reduced it; organizational mechanisms have explained whether those stacks have been operated reliably; and environmental mechanisms have explained why certain optimizations (e.g., approximate answers) have been less adopted in ERP-aligned reporting contexts (Shi et al., 2016). This has complemented IS success measurement perspectives, where system success has been multi-dimensional and mediated through information quality, use, and net impacts rather than purely technical efficiency. The conceptual framework developed in the study—connecting latency, availability, and decision-support quality—has also been theoretically consistent with BI capability and BI success findings that have emphasized the interaction between BI capabilities and decision environments. The discussion has therefore argued that the study has contributed a theory-consistent synthesis structure: the RTBI performance constructs have been operationalizable through a small set of interpretable measures and coding rules (including the L_E2E decomposition and availability A = MTBF/(MTBF+MTTR)), while TOE has explained why those measures have varied across cases (Vassiliadis, 2009). Importantly, the theoretical implication has not been that TOE has replaced technical models, but that it has offered a bridge between technical architecture research and organizational BI value research, a gap that has often appeared in real-time analytics discourse where system designs have been discussed without parallel operational capability analysis. The study has therefore supported an integrated view where "best practice" in real-time SAP retail BI has been theoretically framed as a socio-technical configuration rather than a technology checklist (Vassiliadis et al., 2017). This framing has also clarified why coded evidence has been stronger for upstream integration mechanisms (which require cross-team governance) than for micro-optimizations in query engines (which are often not visible at the organizational decision level): the literature has reported most richly on phenomena that have spanned both technology and organization, aligning naturally with TOE's explanatory scope (Wang et al., 2010).

The limitations revisited have centered on the review's dependence on heterogeneous reporting and on the need to encode narrative evidence into Likert-scaled constructs, which has been methodologically necessary but has constrained inferential certainty (Shute et al., 2013). First, many SAP retail BI publications and case-oriented sources have not reported standardized numeric latency or availability measures, which has limited the ability to compute direct effect sizes or conduct meta-analytic aggregation. This pattern has been consistent with earlier BI success scholarship that has frequently relied on perceptual measures and survey-based indicators because organizations have not always disclosed operational metrics (Nam et al., 2010). Second, the Likert evidence scale has provided a transparent normalization mechanism across sources, but it has depended on coding judgment, even when supported by a codebook and evidence-strength tags. Third, because the study has been cross-sectional and literature-based, the synthesis has not established causality in the experimental sense; it has established repeated associations and convergent explanations across independent studies, which has been appropriate for a qualitative review but has remained a limitation for causal claims. Fourth, the corpus has likely exhibited publication bias toward successful or notable architectures, particularly in practitioner-adjacent reports and vendor-influenced write-ups, which has potentially underrepresented failure cases and negative outcomes (Oliveira et al., 2014). Fifth, the "SAP S/4HANA retail" scope has necessarily included transferable systems research (stream processing, HTAP, reliability engineering) to strengthen mechanism explanations, yet some of these studies have not been SAP-specific, creating a generalizability trade-off: mechanism validity has been strong, while direct SAP-context fidelity has varied. Finally, environmental variability (regional compliance requirements, infrastructure cost constraints, differing retailer scale) has limited one-to-one comparability, even though TOE has been used to structure those differences explicitly. The discussion has therefore treated the results as robust evidence of convergent patterns—especially around CDC/incremental refresh, in-memory serving, and observability + automated recovery—while acknowledging that precise numeric magnitudes have remained dependent on how each source has measured and disclosed performance. These limitations have justified the study's approach of presenting numeric summaries as descriptive evidence of prevalence and strength rather than as definitive benchmarking results, and they have framed the future research agenda toward more standardized reporting and more explicit measurement designs (Popovič et al., 2012).

Future research (FR) has been the most actionable extension of the study, and it has been proposed as a concrete model and research program that has improved measurability, comparability, and theory integration for real-time SAP S/4HANA retail BI. Building on TOE and on the study's RTBI_Index logic, future researchers have been able to develop and validate a TOE-RTBI Performance Maturity Model (TRPMM) that has explicitly combined (1) technology-stack maturity, (2) organizational operations maturity, and (3) environmental constraint alignment into staged capability levels that predict real-time BI outcomes (Mao et al., 2017). The proposed model has been structured as follows: TRPMM Level 1 (Batch BI) has relied on periodic refresh with limited observability; Level 2 (Incremental BI) has introduced CDC/incremental refresh and basic monitoring; Level 3 (Event-Driven BI) has added stateful streaming, automated scaling, and governed semantic layers; Level 4 (Resilient Real-Time BI) has integrated automated failover, standardized recovery semantics, and full observability (lag, tail latency, tracing); and Level 5 (Adaptive Real-Time BI) has added workload-aware orchestration and governed experimentation loops that tie BI outputs to operational interventions. Researchers have been able to operationalize TRPMM with a measurable instrument: a TOE-based survey for organizational/environmental readiness plus direct system telemetry capture for L_E2E components and A, thereby reducing reliance on narrative-only evidence (Nam et al., 2010). A second FR direction has been to validate the study's "pipeline-first" latency interpretation by collecting standardized latency decomposition metrics across multiple SAP retail cases and analyzing which stage has dominated (ingest vs transform vs query), building directly on tail latency insights in distributed systems. A third FR direction has proposed a Recovery Semantics Disclosure Protocol (RSDP) for real-time BI publications: future case studies have reported checkpointing cadence, exactly-once boundaries, sink commit strategies, and replay correctness tests, aligning with stateful stream processing reliability research and improving reproducibility (Mersereau, 2015). A fourth FR direction has been to test when approximate query processing can be safely introduced in SAP retail BI by defining KPI risk classes (audit-critical vs operational-exploratory) and empirically evaluating whether approximate "fast-first" outputs have improved decision speed without degrading trust, thereby extending AQP work into governance-sensitive ERP settings. Finally, future researchers have been able to build a structural model that has tested DSQ as a mediator between technical performance and benefits using mixed methods: telemetry-based latency and availability as exogenous variables, DSQ as a measured mediator, and operational KPIs (stockout reduction, promotion response time) as outcomes, thus integrating IS success theory with systems engineering evidence. Through TRPMM and the associated measurement program, FR has offered a clear path for improving the scientific rigor of real-time SAP S/4HANA retail BI research while remaining aligned with the socio-technical reality captured by TOE (Isik et al., 2013).

CONCLUSION

This research has concluded that advanced computing frameworks have enabled real-time SAP S/4HANA retail business intelligence most effectively when they have been engineered as an integrated, governed, and reliability-aware pipeline rather than as isolated technical upgrades. Across the synthesized evidence base, the dominant architecture pattern has combined CDC or incremental refresh mechanisms with event-driven processing and in-memory or HTAP-style serving, and this hybrid stack has been the most consistently associated with reduced end-to-end latency, improved freshness of analytical views, and more stable decision support for inventory, pricing, promotion, and omnichannel fulfillment workflows. The study has also concluded that latency optimization in SAP-centered retail BI has been primarily achieved through upstream freshness control and incremental maintainability—such as delta-based ingestion, micro-batching, and view maintenance—supported by pushdown computation and workload isolation, because these mechanisms have reduced recomputation costs and have preserved predictable refresh behavior during demand surges. In reliability terms, the research has determined that real-time BI trustworthiness has depended on the co-deployment of observability and automated recovery practices, since metrics/logs/traces, lag monitoring, and incident-aware automation have enabled faster detection of pipeline degradation, more consistent recovery from partial failures, and greater continuity of KPI delivery during peak periods. The conclusions have further shown that reliability has extended beyond service uptime to include semantic stability of KPIs, lineage clarity, and controlled state recovery for streaming aggregates, reflecting the governance-sensitive nature of SAP retail reporting environments. The study has met its objectives by (1) classifying the most frequently adopted framework categories in the corpus, (2) synthesizing the most recurrent processing and latency optimization techniques across pipeline stages, (3) consolidating resilience practices that have supported system continuity and metric correctness, and (4) mapping these patterns to retail decision workflows using a cross-sectional case-based evidence structure supported by Likert-coded quantitative summaries. Hypotheses assessment has reinforced that hybrid stacks have been associated with stronger latency outcomes than single-layer designs, CDC/event-driven integration has been associated with improved freshness compared to batch refresh patterns, and observability combined with automated failover has been associated with stronger reliability outcomes than basic monitoring and manual recovery. Finally, the research has concluded that the TOE theoretical lens has provided a coherent explanation of why technical frameworks have translated into operational BI performance only under certain conditions: technology mechanisms have created potential performance gains, organizational readiness has determined whether those mechanisms have been implemented and operated consistently, and environmental constraints such as auditability and competitive pressure have shaped acceptable latency–reliability trade-offs. Overall, the study has established a consolidated evidence-based understanding that real-time SAP S/4HANA retail BI performance has emerged from a socio-technical configuration in which continuous integration, in-memory analytics, and resilience engineering have been jointly governed to deliver timely and dependable intelligence outputs aligned with retail decision cycles.

RECOMMENDATIONS

The recommendations of this research have emphasized that organizations seeking real-time SAP S/4HANA retail business intelligence have needed to pursue a pipeline-governed, reliability-aware architecture roadmap that has aligned technology choices with organizational capability and environmental constraints. First, retailers have been recommended to prioritize CDC and incremental refresh as the default integration strategy for operational BI workloads because continuous delta propagation has minimized freshness delay, reduced full-load recomputation, and stabilized refresh behavior during peak trading periods; where CDC has not been feasible for specific sources, micro-batching with strict refresh SLAs and delta validation rules has been recommended as the next-best alternative. Second, organizations have been recommended to adopt in-memory or HTAP-aligned serving patterns for high-frequency decision dashboards (inventory exceptions, promotion monitoring, fulfillment visibility), while maintaining a governed analytical layer for historized and finance-sensitive reporting, since the evidence has shown that hybrid serving has balanced low latency with controllership requirements. Third, workload isolation has been recommended as a non-negotiable design principle: BI refresh and heavy analytical workloads have been recommended to be separated from mission-critical transactional workloads through resource quotas, query governance, priority scheduling, and capacity segmentation, thereby protecting both performance and business continuity. Fourth, reliability engineering has been recommended to be institutionalized through a combined program of observability and automated recovery, including standardized telemetry (ingestion lag, checkpoint duration, tail latency percentiles), alerting tied to business-impact thresholds, automated failover and HA/DR runbooks, and deterministic replay procedures for stateful pipelines; this approach has been recommended because "up" systems have not guaranteed correct dashboards if recovery semantics and commit boundaries have remained ambiguous. Fifth, SAP retail BI programs have been recommended to strengthen semantic governance by establishing a controlled KPI dictionary, data lineage documentation, and aggregation rules that have been enforced through a semantic layer or virtual model, since real-time speed has not compensated for inconsistent definitions across regions, channels, and departments. Sixth, organizations have been recommended to build cross-functional operating ownership that has included IT/data engineering, retail operations, finance controllership, and security/compliance teams, because the evidence has indicated that real-time BI performance has depended on organizational readiness as much as on architecture selection; this has included training programs and role definitions for incident response, data quality triage, and KPI stewardship. Seventh, to support continuous improvement, retailers have been recommended to implement a measurement framework that has decomposed end-to-end latency into pipeline components (ingestion, transformation, persistence, query, visualization) and has tracked availability through MTBF and MTTR, enabling objective prioritization of bottlenecks and repeatable reporting of improvements. Finally, as a strategic step, organizations have been recommended to adopt a staged maturity approach aligned with the TOE perspective—starting with incremental integration and basic monitoring, then progressing toward event-driven processing, resilient state management, automated recovery, and workload-aware orchestration—so that investments have been sequenced according to capability readiness and compliance requirements rather than driven by isolated tool adoption.

LIMITATIONS

This study has had several limitations that have reflected both the methodological constraints of a literature review and the evidence characteristics of real-time SAP S/4HANA retail business intelligence research. First, the corpus has included heterogeneous publication types and reporting styles, and many sources have not disclosed standardized performance metrics for end-to-end latency, availability, or throughput; as a result, the study has relied on structured qualitative extraction and a five-point Likert evidence scale to normalize findings across studies, which has enabled comparability but has limited the precision of numeric interpretation and has not supported meta-analytic effect-size estimation. Second, although a codebook and evidence-strength rules have been applied, Likert-based coding has inevitably depended on interpretive judgment when studies have described outcomes narratively rather than with measured values, and this has introduced potential subjectivity even when coding definitions have been consistently applied. Third, the cross-sectional synthesis has captured associations and convergent patterns rather than experimental causality, meaning that supported hypotheses have represented repeated literature-backed relationships between architecture patterns and reported outcomes, not definitive causal proof that specific framework stacks have always caused latency reduction or reliability improvements in every retail context. Fourth, publication bias has likely been present because successful or notable architectures have been more likely to be written up than failed implementations, and practitioner- or vendor-adjacent reports have sometimes emphasized benefits more than trade-offs; although the study has attempted to mitigate this through quality appraisal and evidence-strength tagging, the resulting corpus has still been shaped by what has been publicly documented. Fifth, even within the SAP S/4HANA focus, the included evidence has varied in contextual fidelity: some studies have been directly SAP-centered, while others have been foundational systems research (stream processing, state management, ETL optimization, HTAP) that has provided transferable mechanisms rather than SAP-specific implementation detail; this has strengthened theoretical grounding but has also created a generalization constraint when translating conclusions into SAP-specific operational guidance. Sixth, the retail domain itself has been diverse across geography, scale, channel mix, and regulatory regimes, and many studies have not fully specified environmental constraints such as data residency requirements, auditability expectations, or vendor support models; consequently, the study has not been able to control for environmental variance systematically, even though the TOE framing has been used to interpret it. Seventh, because the research has been literature-based, it has not incorporated direct system telemetry, primary interviews, or organization-level operational data, which has limited the ability to validate coded constructs such as decision-support quality and reliability practices against objective logs. Finally, the study's scope has prioritized advanced computing frameworks and performance outcomes, which has meant that some adjacent determinants of BI success—such as change management, user training depth, and political dynamics of KPI ownership—have not been treated with the same granularity as technical and operational engineering mechanisms. These limitations have indicated that the findings have been most appropriately interpreted as a structured, evidence-based synthesis of dominant patterns and their reported associations, rather than as a definitive benchmark ranking of architectures or a substitute for organization-specific performance testing and governance assessment.

REFERENCES

[1]. Abadi, D. J., Ahmad, Y., Balazinska, M., Cetintemel, U., Cherniack, M., Hwang, J.-H., Lindner, W., Maskey, A. S., Rasin, A., Ryvkina, E., Tatbul, N., Xing, Y., & Zdonik, S. (2005). *The design of the Borealis stream processing engine* Proceedings of the 2nd Biennial Conference on Innovative Data Systems Research (CIDR),

[2]. Abadi, D. J., Madden, S. R., & Hachem, N. (2008). *Column-stores vs. row-stores: How different are they really?* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[3]. Akidau, T., Balikov, A., Bekiroğlu, K., Chernyak, S., Haberman, J., Lax, R., McVeety, S., Mills, D., Nordstrom, P., & Whittle, S. (2013). MillWheel: Fault-tolerant stream processing at internet scale. *Proceedings of the VLDB Endowment*, *6*(11), 1033–1044. https://doi.org/10.14778/2536222.2536229

[4]. Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R. J., Lax, R., McVeety, S., Mills, D., Perry, F., Schmidt, E., & Whittle, S. (2015). The dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. *Proceedings of the VLDB Endowment*, *8*(12), 1792–1803. https://doi.org/10.14778/2824032.2824076

[5]. Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50–58. https://doi.org/10.1145/1721654.1721672

[6]. Armbrust, M., Xin, R. S., Lian, C., Huai, Y., Liu, D., Bradley, J. K., Meng, X., Kaftan, T., Franklin, M. J., Ghodsi, A., & Zaharia, M. (2015). *Spark SQL: Relational data processing in Spark* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[7]. Awa, H. O., & Ojiabo, O. U. (2016). A model of adoption determinants of ERP within T-O-E framework. *Information Technology & People*, *29*(4), 901–930. https://doi.org/10.1108/itp-03-2015-0068

[8]. Bailis, P., Alvaro, P., & Gulwani, S. (2017). Research for practice: Tracing and debugging distributed systems; programming by examples. *Communications of the ACM*, *60*(7), 46–49. https://doi.org/10.1145/3052942

[9]. Banaeianjahromi, N., & Smolander, K. (2016). What do we know about the role of enterprise architecture in enterprise integration? A systematic mapping study. *Journal of Enterprise Information Management*, *29*(1), 140–164. https://doi.org/10.1108/jeim-12-2014-0114

[10]. Biswas, S., Sahu, S., & Sahoo, S. (2020). Incremental loading techniques in data warehousing: A systematic review. *International Journal of Enterprise Information Systems*, *16*(1), 1–20. https://doi.org/10.1007/s11334-019-00344-4

[11]. Bradlow, E. T., Gangwar, M., Kopalle, P., & Voleti, S. (2017). The role of big data and predictive analytics in retailing. *Journal of Retailing*, *93*(1), 79–95. https://doi.org/10.1016/j.jretai.2016.12.004

[12]. Brewer, E. A. (2012). CAP twelve years later: How the "rules" have changed. *Computer*, *45*(2), 23–29. https://doi.org/10.1109/mc.2012.37

[13]. Carbone, P., Ewen, S., Fóra, G., Haridi, S., Richter, S., & Tzoumas, K. (2017). *State management in Apache Flink®: Consistent stateful distributed stream processing* Proceedings of the VLDB Endowment,

[14]. Chen, L., & Mersereau, A. J. (2015). Analytics for operational visibility in the retail store: The cases of censored demand and inventory record inaccuracy. In N. Agrawal & S. A. Smith (Eds.), *Retail supply chain management: Quantitative models and empirical studies* (pp. 79–112). https://doi.org/10.1007/978-1-4899-7562-1_5

[15]. Chou, Y.-C., Hsu, Y.-C., & Chou, C.-C. (2015). Factors of data infrastructure and resource support influencing the integration of business intelligence into enterprise resource planning systems. *International Journal of Innovation and Information Development*, *4*(3), 205–218. https://doi.org/10.1504/ijiids.2015.070822

[16]. Corbett, J. C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J. J., Ghemawat, S., Gubarev, A., Heiser, C., Hochschild, P., Hsieh, W., Kanthak, S., Kogan, E., Li, H., Lloyd, A., Melnik, S., Mwaura, D., Nagle, D., Quinlan, S., . . . Woodford, D. (2013). Spanner: Google's globally distributed database. *ACM Transactions on Computer Systems*, *31*(3), Article 8. https://doi.org/10.1145/2491245

[17]. Cruz-Jesus, F., Pinheiro, A., & Oliveira, T. (2019). Understanding CRM adoption stages: Empirical analysis building on the TOE framework. *Computers in Industry*, *109*, 1–13. https://doi.org/10.1016/j.compind.2019.03.007

[18]. Dean, J., & Barroso, L. A. (2013). The tail at scale. *Communications of the ACM*, *56*(2), 74–80. https://doi.org/10.1145/2408776.2408794

[19]. Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

[20]. DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., & Vogels, W. (2007). *Dynamo: Amazon's highly available key-value store* Proceedings of the ACM SIGOPS Symposium on Operating Systems Principles,

[21]. Di Francesco, P., Lago, P., & Malavolta, I. (2019). Architecting with microservices: A systematic mapping study. *Journal of Systems and Software*, *150*, 77–97. https://doi.org/10.1016/j.jss.2019.01.001

[22]. Diaconu, C., Freedman, C., Ismert, E., Larson, P.-Å., Mittal, P., Stonecipher, R., Verma, N., & Zwilling, M. (2013). *Hekaton: SQL Server's memory-optimized OLTP engine* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[23]. Elbashir, M. Z., Collier, P. A., & Davern, M. J. (2008). Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. *International Journal of Accounting Information Systems*, *9*(3), 135–153. https://doi.org/10.1016/j.accinf.2008.03.001

[24]. Färber, F., Cha, S. K., Primsch, J., Bornhövd, C., Sigg, S., & Lehner, W. (2012). SAP HANA database: Data management for modern business applications. *ACM SIGMOD Record*, *40*(4), 45–51. https://doi.org/10.1145/2094114.2094126

[25]. Faysal, K., & Shamsunnahar, C. (2022). Digital Ledger Optimization Techniques for Enhancing Transaction Speed and Reporting Accuracy in Accounting Systems. *American Journal of Scholarly Research and Innovation*, *1*(02), 171–222. https://doi.org/10.63125/33t06k57

[26]. Gable, G. G., Sedera, D., & Chan, T. (2008). Re-conceptualizing information system success: The IS-Impact measurement model. *Journal of the Association for Information Systems*, *9*(7), 377–408. https://doi.org/10.17705/1jais.00164

[27]. Gailly, F., Geerts, G. L., & Poels, G. (2014). Capturing enterprise data integration challenges using a semiotic data quality framework. *Business & Information Systems Engineering*, *6*(4), 219–231. https://doi.org/10.1007/s12599-014-0365-x

[28]. Ghazanfari, M., Jafari, M., & Rouhani, S. (2011). A tool to evaluate the business intelligence of enterprise systems. *Scientia Iranica*, *18*(6), 1579–1590. https://doi.org/10.1016/j.scient.2011.11.011

[29]. Gupta, M., & George, J. F. (2016). Toward the development of a big data analytics capability. *Information & Management*, *53*(8), 1049–1064. https://doi.org/10.1016/j.im.2016.07.004

[30]. Habibullah, S. M., & Zaheda, K. (2022). Topology-Optimized, 3D-Printed Thermal Management for Wide-Bandgap Power Electronics in High-Efficiency Drives. *Journal of Sustainable Development and Policy*, *1*(02), 134-167. https://doi.org/10.63125/p8m2p864

[31]. Hsu, P.-F., Ray, S., & Li-Hsieh, Y.-Y. (2014). Examining cloud computing adoption intention, pricing mechanism, and deployment model. *International Journal of Information Management*, *34*(4), 474–488. https://doi.org/10.1016/j.ijinfomgt.2014.04.006

[32]. Hübner, A., Kuhn, H., & Wollenburg, J. (2016). Last mile fulfilment and distribution in omni-channel grocery retailing: A strategic planning framework. *International Journal of Retail & Distribution Management*, *44*(3), 228–247. https://doi.org/10.1108/ijrdm-11-2014-0154

[33]. Ibáñez, L.-D., Mazón, J.-N., & Simperl, E. (2017). Towards semantic assessment of summarizability in self-service business intelligence. In *New Trends in Databases and Information Systems* (pp. 174–185). https://doi.org/10.1007/978-3-319-67162-8_18

[34]. Isik, Ö., Jones, M. C., & Sidorova, A. (2011). Business intelligence (BI) success and the role of BI capabilities. *Intelligent Systems in Accounting, Finance and Management*, *18*(4), 161–176. https://doi.org/10.1002/isaf.329

[35]. Isik, Ö., Jones, M. C., & Sidorova, A. (2013). Business intelligence success: The roles of BI capabilities and decision environments. *Information & Management*, *50*(1), 13–23. https://doi.org/10.1016/j.im.2012.12.001

[36]. Jia, L., Yao, X., & Zhang, Y. (2013). Real-time data warehouse architecture and key technologies. In *Database Systems for Advanced Applications* (pp. 441–453). https://doi.org/10.1007/978-3-642-53703-5_35

[37]. Kemper, A., & Neumann, T. (2011). *HyPer: A hybrid OLTP&OLAP main memory database system based on virtual memory snapshots* 2011 IEEE 27th International Conference on Data Engineering,

[38]. Koch, C., Ahmad, Y., Kennedy, O., Nikolic, M., Nötzli, A., Lupei, D., & Shaikhha, A. (2014). DBToaster: Higher-order delta processing for dynamic, frequently fresh views. *The VLDB Journal*, *23*(2), 253–278. https://doi.org/10.1007/s00778-013-0348-4

[39]. Krishnamoorthi, S., & Mathew, S. K. (2010). A case analysis of Savecom: The role of indigenous leadership in implementing a business intelligence system. *International Journal of Information Management*, *30*(4), 368–373. https://doi.org/10.1016/j.ijinfomgt.2010.04.002

[40]. Leis, V., Boncz, P. A., Kemper, A., & Neumann, T. (2014). *Morsel-driven parallelism: A NUMA-aware query evaluation framework for the many-core age* Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data,

[41]. Li, K., & Li, G. (2018). Approximate query processing: What is new and where to go? A survey on approximate query processing. *Data Science and Engineering*, *3*, 379–397. https://doi.org/10.1007/s41019-018-0074-4

[42]. Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., & Czajkowski, G. (2010). *Pregel: A system for large-scale graph processing* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[43]. Malik, S. U. R., Khan, S. U., Ewen, S. J., Tziritas, N., Kolodziej, J., Zomaya, A. Y., Madani, S. A., Min-Allah, N., Wang, L., Xu, C.-Z., Malluhi, Q. M., Pecero, J. E., Balaji, P., Vishnu, A., Ranjan, R., Zeadally, S., & Li, H. (2016). Performance analysis of data intensive cloud systems based on data management and replication: A survey. *Distributed and Parallel Databases*, *34*, 179–215. https://doi.org/10.1007/s10619-015-7173-2

[44]. Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, *19*(4), 2322–2358. https://doi.org/10.1109/comst.2017.2745201

[45]. Md Abubakar Siddique, A., & Md. Al Amin, K. (2022). Data-Driven Ergonomic Risk Analysis Using Wearable Sensor Networks and Deep Learning for Injury Prevention in Industrial Workplaces. *American Journal of Data Science and Analytics*, *3*(06), 01-39. https://doi.org/10.63125/61w9ba54

[46]. Md, F., & Islam, M. D. Z. (2022). Quantitative Risk Modeling of VPN Misconfigurations and Firewall Rule Drift in Hybrid Cloud Networks. *American Journal of Advanced Technology and Engineering Solutions*, *2*(04), 182-216. https://doi.org/10.63125/fa4qdz07

[47]. Md. Mosheur, R., & Rebeka, S. (2021). Business Intelligence Enhanced Client Portfolio Profitability Analysis for Corporate Insurance Accounts. *International Journal of Business and Economics Insights*, *1*(3), 01–36. https://doi.org/10.63125/qcs8d475

[48]. Md. Mosheur, R., & Rebeka, S. (2022). Data-Driven Framework for Service Issue Escalation and Resolution in Large Scale Insurance Portfolios. *Review of Applied Science and Technology*, *1*(04), 216–249. https://doi.org/10.63125/dkzy5k88

[49]. Melnik, S., Gubarev, A., Long, J. J., Romer, G., Shivakumar, S., Tolton, M., & Vassilakis, T. (2010). Dremel: Interactive analysis of web-scale datasets. *Proceedings of the VLDB Endowment*, *3*(1–2), 330–339. https://doi.org/10.14778/1920841.1920886

[50]. Mersereau, A. J. (2015). Demand estimation from censored observations with inventory record inaccuracy. *Manufacturing & Service Operations Management*, *17*(3), 335–349. https://doi.org/10.1287/msom.2015.0520

[51]. Mikalef, P., Krogstie, J., Pappas, I. O., & Giannakos, M. (2018). Big data analytics capabilities: A systematic literature review and research agenda. *Information Systems and e-Business Management*, *16*(3), 547–578. https://doi.org/10.1007/s10257-017-0362-y

[52]. Mikalef, P., Krogstie, J., Pappas, I. O., & Pavlou, P. (2019). Exploring the relationship between big data analytics capability and competitive performance: The mediating roles of dynamic and operational capabilities. *Information & Management*. https://doi.org/10.1016/j.im.2019.05.004

[53]. Mostafa, K., & Md Tohidul, I. (2022). A Quantitative Financial Impact Assessment of Digital Trade Platforms on Export Performance, Capital Efficiency, and Market Competitiveness. *Journal of Sustainable Development and Policy*, *1*(03), 01-26. https://doi.org/10.63125/pt5v9517

[54]. Mudzana, T., & Maharaj, M. (2015). Measuring the success of business-intelligence systems in South Africa: An empirical investigation applying the DeLone and McLean Model. *South African Journal of Information Management*, *17*(1), Article 646. https://doi.org/10.4102/sajim.v17i1.646

[55]. Nam, B., Shin, M., Andrade, H., & Sussman, A. (2010). Multiple query scheduling for distributed semantic caches. *Journal of Parallel and Distributed Computing*, *70*(5), 598–611. https://doi.org/10.1016/j.jpdc.2010.02.002

[56]. Neumann, T. (2011). *Efficiently compiling efficient query plans for modern hardware* Proceedings of the VLDB Endowment,

[57]. Neumann, T., & Moerkotte, G. (2011). *Characteristic sets: Accurate cardinality estimation for RDF queries with multiple joins* 2011 IEEE 27th International Conference on Data Engineering,

[58]. Nofal, M. I., & Yusof, Z. M. (2013). Integration of business intelligence and enterprise resource planning within organizations. *Procedia Technology*, *11*, 658–665. https://doi.org/10.1016/j.protcy.2013.12.242

[59]. Oliveira, T., Thomas, M., & Espadanal, M. (2014). Assessing the determinants of cloud computing adoption: An analysis of the manufacturing and services sectors. *Information & Management*, *51*(5), 497–510. https://doi.org/10.1016/j.im.2014.03.006

[60]. Olston, C., Reed, B., Srivastava, U., Kumar, R., & Tomkins, A. (2008). *Pig Latin: A not-so-foreign language for data processing* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[61]. Petter, S., DeLone, W., & McLean, E. R. (2008). Measuring information systems success: Models, dimensions, measures, and interrelationships. *European Journal of Information Systems*, *17*(3), 236–263. https://doi.org/10.1057/ejis.2008.15

[62]. Plattner, H. (2009). *A common database approach for OLTP and OLAP using an in-memory column database* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[63]. Popovič, A., Hackney, R., Coelho, P. S., & Jaklič, J. (2012). Towards business intelligence systems success: Effects of maturity and culture on analytical decision making. *Decision Support Systems*, *54*(1), 729–739. https://doi.org/10.1016/j.dss.2012.10.001

[64]. Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/jiot.2016.2579198

[65]. Shute, J., Oancea, M., Ellner, S., Handy, B., Rollins, E., & Vingralek, R. (2013). F1: A distributed SQL database that scales. *Proceedings of the VLDB Endowment*, *6*(11), 1068–1079. https://doi.org/10.14778/2536222.2536232

[66]. Sikka, V., Färber, F., Lehner, W., Cha, S. K., Peh, T., & Bornhövd, C. (2012). *Efficient transaction processing in SAP HANA database: The end of a column store myth* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[67]. Simitsis, A., Vassiliadis, P., & Sellis, T. K. (2005). State-space optimization of ETL workflows. *IEEE Transactions on Knowledge and Data Engineering*, *17*(10), 1404–1419. https://doi.org/10.1109/tkde.2005.169

[68]. Thomson, A., Diamond, T., Weng, S.-C., Ren, K., Shao, P., & Abadi, D. J. (2012). *Calvin: Fast distributed transactions for partitioned database systems* Proceedings of the ACM SIGMOD International Conference on Management of Data,

[69]. To, Q.-C., Soto, J., & Markl, V. (2018). A survey of state management in big data processing systems. *The VLDB Journal*, *27*, 847–872. https://doi.org/10.1007/s00778-018-0514-9

[70]. Torres, R., & Sidorova, A. (2019). Reconceptualizing information quality as effective use in the context of business intelligence and analytics. *International Journal of Information Management*, *49*, 316–329. https://doi.org/10.1016/j.ijinfomgt.2019.05.028

[71]. Torres, R., Sidorova, A., & Jones, M. C. (2018). Enabling firm performance through business intelligence and analytics: A dynamic capabilities perspective. *Information & Management*, *55*(7), 822–839. https://doi.org/10.1016/j.im.2018.03.010

[72]. Varghese, B., Wang, N., Barbhuiya, S., Kilpatrick, P., & Nikolopoulos, D. S. (2016). *Challenges and opportunities in edge computing* Proceedings of the IEEE International Conference on Smart Cloud (SmartCloud 2016),

[73]. Vassiliadis, P. (2009). A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining*, *5*(3), 1–27. https://doi.org/10.4018/jdwm.2009070101

[74]. Vassiliadis, P., Simitsis, A., & Skiadopoulos, S. (2017). From conceptual design to performance optimization of ETL workflows: Current state of research and open problems. *The VLDB Journal*, *26*, 777–801. https://doi.org/10.1007/s00778-017-0477-2

[75]. Verhoef, P. C., Kannan, P. K., & Inman, J. J. (2015). From multi-channel retailing to omni-channel retailing: Introduction to the special issue on multi-channel retailing. *Journal of Retailing*, *91*(2), 174–181. https://doi.org/10.1016/j.jretai.2015.02.005

[76]. Verma, A., Pedrosa, L., Korupolu, M., Oppenheimer, D., Tune, E., & Wilkes, J. (2015). *Large-scale cluster management at Google with Borg* Proceedings of the European Conference on Computer Systems (EuroSys),

[77]. Vogels, W. (2009). Eventually consistent. *Communications of the ACM*, *52*(1), 40–44. https://doi.org/10.1145/1435417.1435432

[78]. Wamba, S. F., Gunasekaran, A., Akter, S., Ren, S. J.-F., Dubey, R., & Childe, S. J. (2017). Big data analytics and firm performance: Effects of dynamic capabilities. *Journal of Business Research*, *70*, 356–365. https://doi.org/10.1016/j.jbusres.2016.08.009

[79]. Wang, Y.-M., Wang, Y.-S., & Yang, Y.-F. (2010). Understanding the determinants of RFID adoption in the manufacturing industry. *Technological Forecasting and Social Change*, *77*(5), 803–815. https://doi.org/10.1016/j.techfore.2010.03.006

[80]. Wieder, B., & Ossimitz, M.-L. (2015). The impact of business intelligence on the quality of decision making—A mediation model. *Procedia Computer Science*, *64*, 1163–1171. https://doi.org/10.1016/j.procs.2015.08.599

[81]. Wixom, B. H. (2010). The BI-based organization. *International Journal of Business Intelligence Research*, *1*(1), 13–28. https://doi.org/10.4018/jbir.2010071702

[82]. Yang, Y., Golab, L., & Özsu, M. T. (2017). ViewDF: Declarative incremental view maintenance for streaming data. *Information Systems*, *71*, 55–67. https://doi.org/10.1016/j.is.2017.07.002

[83]. Zvara, Z., Szabó, P. G. N., Balázs, B., & Benczúr, A. (2018). Optimizing distributed data stream processing by tracing. *Future Generation Computer Systems*, *89*, 468–480. https://doi.org/10.1016/j.future.2018.06.047