

(REVIEW ARTICLE)



SCALABLE LOYALTY SETTLEMENT ARCHITECTURE FOR TRANSACTION INTEGRITY IN DISTRIBUTED RETAIL NETWORKS

Swadeep Vendoti *

JNTU Hyderabad

* Corresponding Author

ORCID Details

Swadeep Vendoti: <https://orcid.org/0009-0009-5079-7429>

World Journal of Advanced Engineering Technology and Sciences, 2026, 19(02), 345–358

Publication history: Received on 24 March 2026; revised on 11 May 2026; accepted on 13 May 2026

Article DOI: <https://doi.org/10.30574/wjaets.2026.19.2.0249>

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

The power of earning and tracking loyalty value across geographies, store apps, and platforms is high, but velocity is precarious and costly. As the number of nodes increases, transaction difficulty increases. Centralized settlement systems cannot effectively maintain transaction integrity, and settlement issues such as double accrual, reconciliation drift, delayed settlement, and disputes further undermine customer trust. This paper presents a novel concept, the Scalable Loyalty Settlement Architecture (SLSA), that ensures transactions within distributed retail networks are not compromised while providing high throughput and low latency. It features a tokenized identity layer, an idempotent accrual engine, an orchestration unit for the settlement process to minimize the number of rounds required to achieve consensus using a quorum, and an append-only integrity ledger verified with a Merkle Tree. We introduce a Transaction Integrity Framework that abstracts capture, validation, deduplication, cryptographic commitment, consensus, and anchoring evidence on the ledger into stages that can be composed, with each stage producing verifiable proofs. Achieved by horizontally scaling the system, but not at the expense of consistency guarantees, via deterministic sharding. In a hypothetical multi-region retail society, experimental testing shows that settlement throughput is significantly higher than centralized and naive distributed baselines, reconciliation loss rates are lower, and disputes are resolved faster. Consistency/Availability, Complexity/Operations, Governance. We believe it's possible and cost-effective to create horizontally scalable settlements that prioritize integrity for large retail.

Keywords: Loyalty Settlement; Transaction Integrity; Distributed Retail Networks; Scalable Architecture; Consensus and Reconciliation.

1 INTRODUCTION

Loyalty settlement is the process customers go through, where all the various entities in a retail network (including the customer) keep a record of the sales made with them in the retail network, count loyalty points provided or earned/redeemed by customers, verify the eligibility of the customer for a loyalty program, and provide that value or matching to other internet service providers in the network with which they will have a financial relationship. Unlike traditional payment transactions, loyalty settlement operations are not confined to a single processing event. Instead, they encompass a series of interconnected activities that occur before, during, and after the initial transaction. These

activities include transaction authorization, reward calculation, customer account updates, partner reconciliation, redemption validation, and audit logging. As a result, loyalty settlement systems require continuous coordination across multiple stakeholders and processing stages to ensure accuracy, consistency, and accountability throughout the transaction lifecycle [1]. They are carried out across geographically distributed infrastructures, such as points of sale, e-commerce solutions, mobile phone applications, self-service kiosks, cloud-based applications, and third-party partner applications, with varying environments, infrastructure, and processing power. This means that state transitions that may never reach the recipient due to retries, simultaneous receipt, disconnected networks, or different clock speeds must occur in the proper order. This is based on the principle that it would be unfair to give a unit of loyalty to more than one person and it would be unfair to withdraw it from one person and give it to a different one, it would be unfair to give someone a partial unit (or pay them for it), or to alter it without permission, or have it disappear if the individual it is for dies or leaves the group [2, 3]. Under these guarantees, several key technologies are incorporated. The first is Distributed Transaction Processing (DTP) Technology, used to maintain the consistency of transactional data. Cryptographic shrewdness and commitment machines are an imperative second technology that can enable the unforgeable “proof” of transactional information. The third is the Append-Only Ledger (AOL) technology, which can create an unchangeable history of all value transactions. If transactions are lost due to network problems or if a transaction is duplicated during settlement, the only way to prevent duplication is to make transactions idempotent so that subsequent retries of the same transaction are effective without data loss. Idempotent transport mechanisms ensure that transactions can be safely retransmitted following network failures, delays, or duplicate message deliveries without causing duplicate processing or inconsistent settlement records. In parallel, deterministic processing guarantees that identical inputs produce identical settlement outcomes across all participating nodes, thereby ensuring consistency, reproducibility, and agreement throughout the distributed settlement network. Moreover, many techniques for deterministic sharding and partition-aware processing from large-scale distributed databases can be used, maintaining per-customer consistency while loyalty balances are processed independently on multiple nodes, enabling horizontal scalability [4, 5]. These all form the technical foundations of a secure, verifiable, and scalable way to settle loyalty and give architects the building blocks for why they might find a centralized or naive distributed implementation difficult, especially as the number of partner ecosystems and transaction volumes proliferate in the retail sector.

The Enterprise incentive reward systems currently used are mainly batch reconciliation reward systems, in which all of the enterprise's daily transactions are accumulated in a point-of-sale system (POS). Then the reward would be sent to the central system for batch settlement processing later, which would issue the individual transaction afterward [6]. This can provide greater simplicity, consistency, and manageability by having one source of truth for reconciliation. But it can come with challenges around runtime performance and scalability, especially given the volume of transactions and additional partner ecosystems. The most straightforward constraint is the settlement latency, the interval between the customer buying the product and subsequent cycles of the system's components and organization, reflecting their counting of the customer as a new customer [7]. In centralized processing, each and every transaction has to be validated, processed, and reconciled by a single settlement engine, which is another potential source of a bottleneck, and over time, batch windows have increased with more outlets, digital channels, and coalition partners joining the network. Also, it is usually done as transaction balancing rather than transaction validation at capture, which means that when a transaction is being replayed, captured transactions that are duplicates, existing transactions that aren't synchronized, or out-of-step updates won't be caught until the transaction is settled, and more resources, like time for manual steps, are needed by the involved parties. The blockchain-based loyalty system applies blockchain technology to provide the immutability and transparency it needs, but is often associated with throughput restrictions, high consensus overhead, and settlement latency, which also limit its applicability to highly scaled retail transaction scenarios where thousands of transactions/second are common [8-10]. More importantly, the currently available solutions focus on the optimization of one element of the design, respectively, scalability, security, and/or operational efficiency. The challenge then is whether consistent distribution of the settlement framework can be developed that will include integrity right from the beginning of the capture of each transaction, the deterministic processing, distributed coordination, and the cryptographic verification and checking of transaction integrity, which will allow high throughput, low settlement latency, fault tolerance and independent checking of transaction integrity for a geographically distributed and not fully trusted retail network.

To address this identified research gap, the paper proposes and tests a Scalable Loyalty Settlement Architecture (SLSA) that integrates transaction integrity into the settlement process rather than relying on post-processing reconciliation. For large-scale retail environments that demand correctness and auditability, the proposed solution is a cross between deterministic transactions, tokenized identity management, distributed sharding, and cryptographic commitment and quorum-based coordination. But SLSA brings integrity to systems by ensuring that the steps for creating and redeeming loyalty value occur at least once, unlike “centralized” or “naive” distributed systems. The architecture also adds a formal Transaction Integrity Framework that specifies settlement as a series of independently verifiable steps and requires that failures, retries, or disputes be detected and resolved deterministically, without compromising system consistency. The study shows that the proposed framework performs much better in terms of throughput, reconciliation accuracy, and efficiency in the dispute-resolution process, and also provides horizontal scalability, compared to several

centralized and distributed baselines. In addition to the technical aspects, the study explores the business implications of integrity-first settlement for the loyalty of nuts-and-bolts companies, including process scalability, proof of integrity, cost savings, consumer trust, ease of compliance, and secure cross-brand collaboration. In order to give a blueprint to researchers and practitioners for developing new distributed loyalty settlement platforms.

The crucial message(s) in this paper are as follows:

- **Foundational Integrity Model:** Provides guidelines for more sophisticated distributed loyalty settlement systems and outlines some common failure modes that may present an integrity challenge, such as Duplicate accrual, replay attack, unclear balances, uncertainty about accrual-match, etc.
- **Scalable Layered Architecture:** proposes a settlement architecture that is horizontally scalable by using a shard-partitioned architecture, an idempotent settlement accrual engine based on determinism, an immutable integrity ledger, and a settlement orchestrator using a quorum system.
- **Transaction Integrity Framework:** This is a multi-step process that operates from transaction captures to verifications, deduplications, cryptographic commit, distributed consensus, and ledger anchoring in a composable and independently verifiable settlement process.
- **Experimental Performance Evaluation:** Validates the anticipated architecture against centralized batch architecture (homogeneous distributed baseline architecture) and performance of the two architectures compared with regard to various performance metrics such as protocol throughput, settlement latency, reconciliation error rate, horizontal scalability time, etc., and at the same time takes into account the disputed number of transactions.
- **Technical and Managerial implications:** Offers practical insights for Engineering teams and Retail decision makers on how to achieve Integrity-First settlement architectures that can help optimize the operation, audit, scale, and multi-partner customer base collaboration.

2 BACKGROUND WORK

Table 1: Summary of Representative Background Studies

Ref.	Research Area / Study	Primary Focus	Core Technique / Methodology	Strengths	Reported Limitation / Research Gap
[8]	Batch Settlement Reconciliation Systems	Operational efficiency	Centralized nightly reconciliation and database recomputation	Simple implementation and centralized control	Delayed integrity verification, high reconciliation overhead, and slow customer updates
[9]	Stream-Based Ledger Anchoring	Low-latency transaction processing	Event sourcing with continuous ledger synchronization	Supports real-time processing and incremental updates	Limited support for dispute resolution and long-term settlement verification
[10]	Distributed Event Processing for Retail Transactions	Real-time scalability	Event-driven messaging and asynchronous processing	Improves responsiveness and distributed processing capacity	Does not guarantee deterministic settlement consistency across nodes
[11]	Partitioned State Management for OLTP Systems	Horizontal scalability	Deterministic sharding and partition-aware transaction routing	Enables parallel execution and high transaction throughput	Weak cross-shard consistency and complex distributed coordination
[12]	Exactly-Once Payment Processing Frameworks	Transaction integrity	Idempotency keys and deterministic processing semantics	Eliminates duplicate processing and replay errors	Relies on centralized coordination that limits scalability
[13]	Tamper-Evident Audit Log Architectures	Verifiability and auditability	Append-only logs with Merkle-tree accumulators	Provides lightweight cryptographic proof generation and tamper detection	Focuses on verification without defining a complete settlement workflow

[14]	Coalition Loyalty Reconciliation Models	Multi-party trust and interoperability	Periodic partner reconciliation and net settlement mechanisms	Enables cross-brand loyalty ecosystems	High reconciliation latency, manual dispute handling, and delayed partner settlement
[15]	Blockchain-Based Loyalty Token Platforms	Decentralized integrity	Consensus-driven distributed ledger technology	Strong immutability, transparency, and trustless verification	Limited throughput, increased consensus latency, and high operational overhead for large-scale retail environments

As a result, scalable loyalty settlement has become an interdisciplinary field, combining ideas from distributed databases, financial transaction systems, event-driven systems, and applied cryptography to address the ever-increasing complexity of today's retail environments. The first-generation loyalty management systems were based on a batch reconciliation concept that was easy to implement and reduced the likelihood of bank settlement issues, but they still had reconciliation issues and limited scalability in terms of transaction volume and timeliness. To address these drawbacks, stream-processing and event-sourcing techniques were adopted as new solutions to ensure near-real-time settlement and continuous sync of the world state, but the latter could be constrained by the mechanisms used to resolve disputes and verify integrity in the long run [9]. Several concepts from distributed database research have been adapted to SQL-based systems. These include sharding, which divides data into independent partitions identified by unique keys [11]. Another key concept is partition-aware processing, where operations within one partition can be executed independently without affecting others. Horizontal scaling further improves throughput by distributing data and workloads across multiple processors. In financial payment systems, preventing duplicate transactions and ensuring settlement correctness are essential requirements. Achieving these objectives necessitates idempotent transaction processing and exactly-once execution guarantees, allowing reliable transaction handling in the presence of retries, communication failures, or network disruptions [12]. The use of an append-only log, Merkle-tree accumulators, and lightweight verifiers to provide tamper evidence without the expense of fully decentralized public blockchains has also improved transaction integrity [13]. The primary focus of multi-party coalition loyalty studies is on safe value exchange between independent parties, but they often rely on periodic reconciliation models, which tie up time and cost in dispute resolution [14]. Throughput problems and consensus latency still hinder the current large-scale application of blockchain-based loyalty programs, as applications need to improve transparency and trust between users and the blockchain [15]. While individual research focuses on a specific aspect of the loyalty settlement, solutions generally optimize scalability, integrity, or operational efficiency while leaving other aspects unoptimized. This void is driving the evolution of an integrated architecture that combines deterministic processing, distributed scalability, cryptographic verification, and effective settlement coordination within a single loyalty settlement environment.

3 PROPOSED SCALABLE LOYALTY SETTLEMENT ARCHITECTURE

The proposed Scalable Loyalty Settlement Architecture aims to overcome shortcomings in centralized settlement processing by integrating Data partitioning (horizontally scalable), Deterministic computation, Cryptographic integrity verification, and distributed transaction management into a single architecture that provides a unified settlement process. The architecture represents each interaction with the customer as a transaction with a verifiable state transition, validated, executed, and permanently recorded in an independent yet coordinated service layer, rather than as a reconciliation chore. Each layer performs a particular task, so that the tasks of transaction capture, identity management, calculating loyalty, coordinating settlement, ensuring integrity, and analytical processing evolve separately without affecting the rest of the system. Loyalty value generated from retail transactions is first encountered within a secure POS validation boundary, where authenticity and compliance are determined before loyalty value is built. Tokenization can abstract identities and store a customer's sensitive details separately from their financial transactions, without sacrificing the ability to use partition keys deterministically to rapidly scale horizontally as needs grow. The architecture also features a reusable loyalty acquisition mechanism that ensures identical transactions always yield identical settlement outcomes, preventing the acquisition of identical rewards, since they are never repeated on retries or due to network errors. A settlement orchestrator for a quorum-based settlement process coordinates value transfers among distributed nodes and ensures that all parties agree on committed state changes before balances are settled. Each committed operation is, all the while, written to an append-only integrity ledger with cryptographic chaining and periodic Merkle root anchoring to verify tamper-evident status and facilitate audits. These are supported by a sharded state store that distributes customer balances across multiple processing partitions, with a reconciliation service that continuously identifies and resolves inconsistencies across these shards. An analytics and monitoring plane operationally consumes immutable records without impacting transactional performance, for

dashboards, dispute management, compliance reporting, and business intelligence. This architecture provides linear scalability and guarantees a predictable settlement pattern, and can handle millions of concurrent transactions while maintaining high availability, low latency, and verifiable settlement consistency across both geographically distributed environments and retail loyalty ecosystems. Figure 1, layered view of the proposed Scalable Loyalty Settlement Architecture (SLSA). Transactions flow from distributed retail channels through validation, tokenization, deterministic loyalty accrual, quorum-based settlement orchestration, and immutable ledger recording. The architecture also incorporates distributed sharded storage and analytics functions. Cross-cutting integrity services ensure secure, scalable, and verifiable loyalty settlement operations.

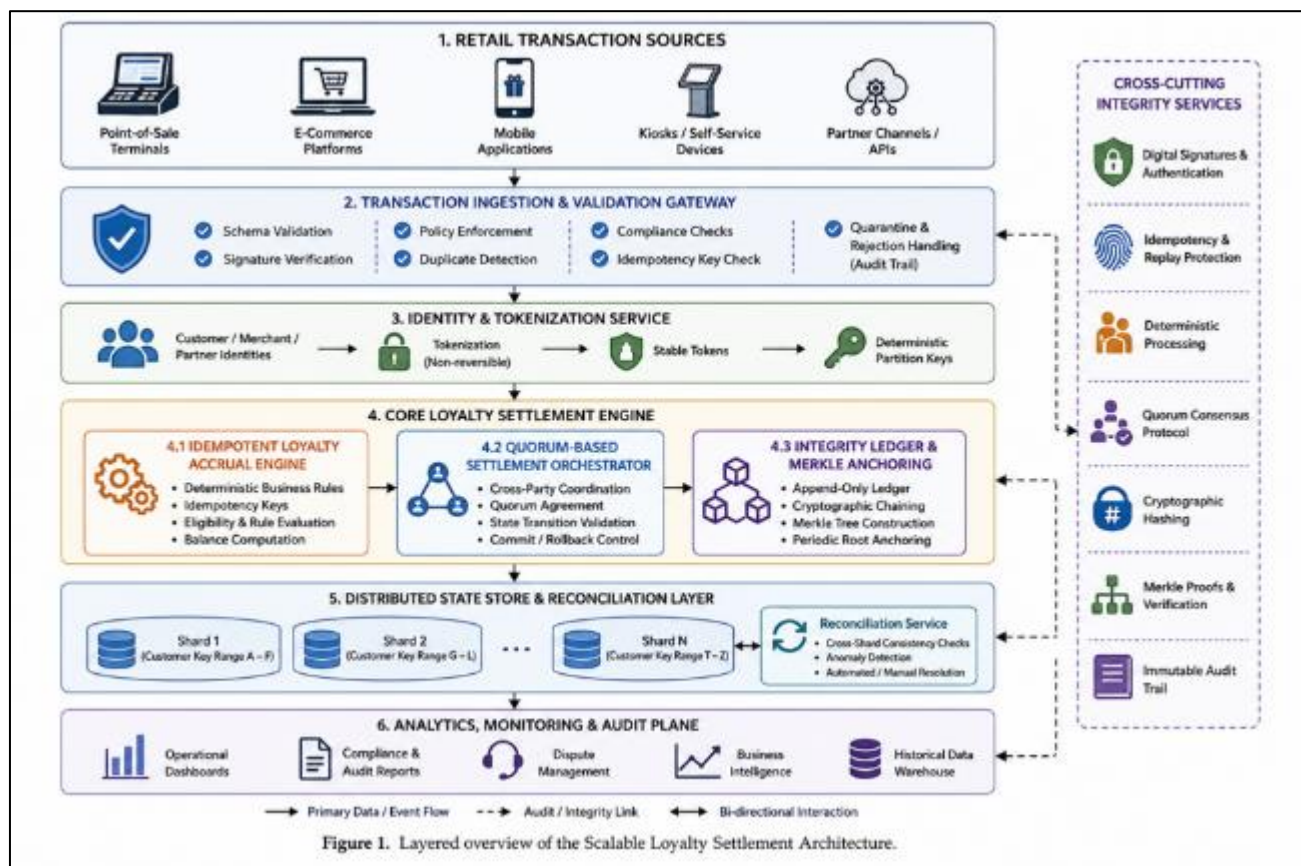


Figure 1: Layered overview of the Scalable Loyalty Settlement Architecture.

3.1 Transaction Ingestion and Validation Gateway

All transactions, irrespective of the loyalty transaction type, pass through the "gate" of the Transaction Ingestion and Validation Gateway for processing and settlement. A large number of checks are performed on all events collected from retail locations, e-commerce sites, partner stores, kiosks, and mobile store/app events to determine their suitability for settlement. Cryptographic signature validation is performed to ensure that the field data, timestamp, transaction identifier, and loyalty attribute(s) included in the required fields are valid according to a predefined schema. Structural schema validation: It ensures that the data included in the required fields, the timestamp, the transaction identifier, and the loyalty attribute(s) are valid according to a predefined schema. The gateway also verifies business policies, checks for duplicate transactions, validates conformant transactions, and validates idempotency keys to ensure that only trusted transactions are passed further in the architecture. Entities that are not found to meet some validation rule requirements are not silently rejected; they are stored in a standardized manner in a quarantined repository for further action, such as operational analysis, dispute resolution, and contingent action. Processing efficiency, reconciliation costs, and settlement-service reputation in the distributed settlement services pipeline depend only on trusted inputs; i.e., bad/mis-formed inputs are filtered away at the boundary of the settlement service, helping to keep costs low.

3.2 Identity and Tokenization Service

The Identity and Tokenization Service separates customer information from loyalty settlement data, reducing the need to store sensitive information within the settlement system. It generates stable and deterministic tokens and manages identities throughout the transaction lifecycle. Customer IDs, merchant accounts, and partner identifiers are converted into representations that cannot be recovered without compromising integrity and without disclosing personally identifiable or commercially sensitive information. This isolation not only provides privacy protection but is also

necessary to comply with regulations and reduce harm from any data breach, as the settlement pieces relate only to unknown tokens rather than real identities. When partitioning distributed storage and distributed processing, each identical identity will retain its identity and map to the same token, thereby guaranteeing a consistent key for the partition. Thus, during normal accrual operations, the need to communicate and synchronize across shards will be reduced, and loyalty will be stored locally within each shard, eliminating the need to synchronize across a defined customer/merchant. The tokenization layer enables efficient coordination of value transfer across programs and between organizations, while keeping cross-program coordination within privacy bounds and providing an optimal foundation for scalable, interoperable loyalty ecosystems with high security and privacy compliance.

3.3 Idempotent Loyalty Accrual Engine

The Idempotent Loyalty Accrual Engine is responsible for calculating the loyalty value accrued on each validated transaction and ensuring that loyalty is not accrued twice for the same transaction. The idempotency key is a globally unique key that prevents new transactions from interpreting each other as transactions, thereby computing settlement results idempotently and making them permanent. One of the pre-checks of the engine is whether the key is already processed or not - if it is, the value from when it was previously processed will be returned instead of calculating the value with this key. This feature prevents double accrual due to message retries, distributed delivery issues, or network failures. Loyalty rules are simply deterministic functions that depend on trusted transaction information and fixed program policies, and that will always yield the same results, irrespective of location or time of processing. This deterministic computation allows execution across multiple nodes and provides consistency in the distributed environment. The engine becomes a sound foundation, therefore, for horizontal scaling, replication, verification, recovery, and many more, from where there is recovery of the operational value: rather than the inflation of rewards that happens in traditional at-least-once architectures.

3.4 Quorum-Based Settlement Orchestrator

Quorum-Based Settlement Orchestrator is a sequence of well-defined states that affects the movement of loyalty value between customers, merchants, brands, and partner organizations. It can no longer be the orchestrator who is responsible for the central node; instead, each participatory node will send the collected payout to the orchestrator, and the orchestrator will execute the settlement transaction only once a determined number of nodes (quorum) have done so. Single-party loyalty accruals have a direct execution path, and only settlement from cross-brands/cross-shards will lead to a distributed protocol in which updates are synchronized in parallel across different processing domains. Majority-based confirmation offers no partial or conflicting balance in the event of temporary node failure or even a network partition, this may affect any party's perception of settlement and even balance. The orchestrator also maintains transaction lifecycle information, failure-to-failure instances, failure-to-failure policies, timers, and recovery information, enabling the resumption of interrupted operations without consistency constraints. The orchestration layer also guarantees that all replicas are in sync before commits, providing guaranteed settlement semantics regardless of the environment, even under high transaction loads, and makes it resistant to network failures, enabling large decentralized retail loyalty ecosystems with almost constant availability.

3.5 Append-Only Integrity Ledger

The Append-only Integrity Ledger (AoIL) is an immutable source of truth in the loyalty settlement system, creating a permanent, chronological Black Book that records every event in the system and cannot be modified. When changes or deletions are made, it's easy to determine, as each transaction is "coded" and "chained" to the one prior. Periodic summaries of bundles of committed transactions adopt the same mechanism, but the transaction committed up to this point is summarized by constructing a Merkle tree as well, and the Merkle root is published as a small cryptographic summary (or "integrity checksum") of the entire bundle of transactions. A lightning inclusion proof that is derived from a published root (without having to replay the entire history of a blockchain) can be checked on any node at any time during an audit and/or investigation into a dispute to verify that an intended block settlement record exists (were it to be compromised, the integrity would be verified by that same method). This introduces a much greater degree of complexity in the verifications while retaining a much more robust tamper-evident property. Operational transparency, regulatory compliance, and partner trust are significantly improved, as they cannot be altered retroactively in terms of settlement history. The ledger offers blockchain-based assurances of integrity and lacks the throughput limitations of a fully distributed consensus network, so it works better for large-volume retail loyalty settlement.

4 TRANSACTION INTEGRITY FRAMEWORK

The Transaction Integrity Framework encompasses the set of mechanisms and guarantees required to ensure that loyalty settlement operations remain accurate, auditable, and resilient in large-scale distributed environments. Rather than treating a loyalty transaction as a single database update, the framework views it as a sequence of interdependent operations that must be executed consistently and verified across the system. This approach helps maintain data integrity, prevent inconsistencies, and ensure reliable settlement outcomes throughout the transaction lifecycle. The

stages each add one level of integrity and provide cryptographic and operational evidence, with no reliance on implicit trust from any single person. The three fundamental invariants are: conservation gives assurance that value will be added or taken away from a loyalty only following approved rules of accrual and redemption respectively; uniqueness provides surety that any transfer of value to or from an account corresponds to a single action, no matter how many you undertake, either deliberately or unwittingly, would there be a network failure or a server-side failure, or would the network service providers each horizontally execute a transaction or attempt to debit credit from the account, it would impact the account only once; and verifiability ensures that we leave something behind that lets us reconstruct and audit any step any account takes. The integrity pipeline starts with the originator retail device, which signs the transaction's provenance. An event back to business rule validation checks, rejection of malformed and unauthorized transactions from settlement processing, through policy enforcement, and finally to event validations. A dedupe stage that is committed points to previously recorded identifiers to prevent retransmissions while maintaining determinism. Successful events are then passed on for deterministic accrual processing (loyalty outcome or computation), followed by cryptographic commitment of the results via hash generation. It not only requires quorum approval for all state transitions to be valid, but also requires that this approval be obtained after certain multi-party transactions. Finally, all dedicated operations are as soon as included in the unchangeable integrity ledger and incorporated into periodically generated "Merkle roots" that provide short proof of membership. The verification artifact may be different for each stage; hence, the gradual emergence of integrity, local failure detection, and gradual recovery, fault diagnosis, and continuous checking throughout the entire chain of loyalty settlement.

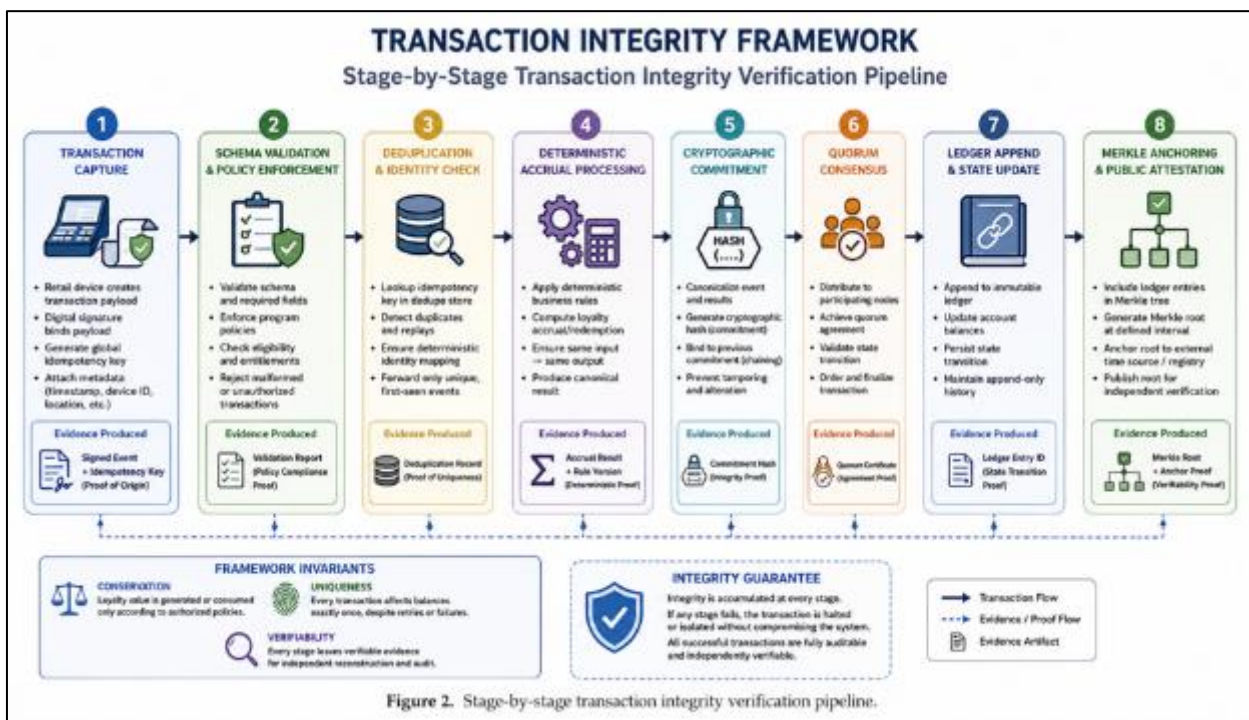


Figure 2: Stage-by-stage transaction integrity verification pipeline.

The Transaction Integrity Framework also sets out a very comprehensive settlement state machine that dictates the Behavior of transactions when they encounter a conflict, a communication breakdown, a duplicate request, or a settlement dispute. Transitions from one state of a transaction to another are explicitly specified, and only the permitted states are states that a transaction may reach, so that when transactions are executed as distributed transactions, there is no ambiguity in what part of the transaction the participant is at or what the permitted states for the transaction are. If it is accepted as valid by the validation gateway (i.e., structurally sound and trusted), it will be in the PENDING state, indicating that the transaction has been validated as correct but has not been processed for settlement. Accrual computation and policy evaluation become deterministic; its transaction is placed into the VALIDATED state in which the "loyalty value is calculated deterministically," and it also satisfies all program requirements. The transaction is not added to the DISPUTED state for any of the above reasons; rather, it changes state. The transaction is not committed too early, and its state changes as a result of any of the above. At this stage, evidence to be submitted to automatic or administrative adjudication can be cryptographic ledger evidence, historical commitments, and quorum evidence. Once they're able to obtain the proper distributed consensus (the COMMITTED state), the transactions get upgraded, and outcomes are final and permanently added to the append-only integrity ledger. Transactions that do not meet consistency requirements and/or policy requirements end up in the REVERSED state and are entered in the ledger as first-class transactions, but don't alter or delete historical transactions. Once reconciliation is successful among all nodes

involved, the transaction moves to the terminal "SETTLED" state, meaning the value cannot be reversed and is independently verifiable. Pertinent transitions, disputes/reversals, and other such transitions are permanently saved, and auditors/operators can rewind the entire chain of such transitions to reach one and only one valid settlement outcome.

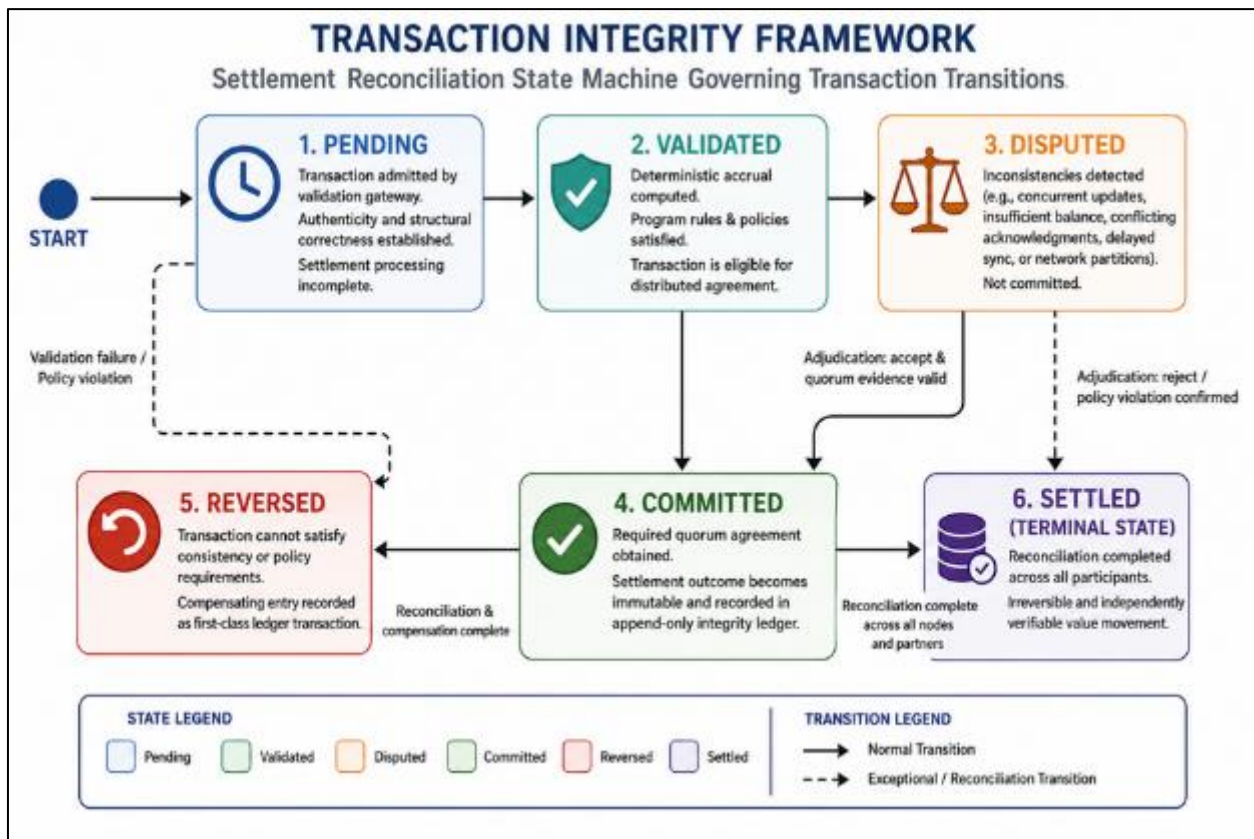


Figure 3: Settlement reconciliation state machine governing transaction transitions.

The multi-stage verification pipeline, combined with the settlement state machine, provides defense-in-depth (DiD) integrity and enhanced security guarantees, and distributes the system as it is used today. Signatures, identity verifications, and idempotence enforcement are ways of ensuring integrity: preventing events from entering the settlement domain that are unauthorized or duplicates. Deterministic (accumulation) algorithms, cryptographic commitments, and quorum-based agreement protocols ensure that all clients, regardless of their processing location, will always receive the same transaction result for the same input. Each settlement can be checked independently after final settlement to verify it is correct, without having to place trust in the previous processing nodes, thanks to the immutability of the ledger, Merkle proofs, and append-only recording. The layered design is explicit: no matter how secure one part is, this trust cannot pass through the layers to the bottom, where the foundation is being breached, breaking the others' guarantees, and it will get easier. Moreover, it provides only one way to compute consensus and anchoring cryptographically across shards and organizations, leaving transactions involving only one party to follow a much simpler path to minimize overhead. This framework, therefore, removes the traditional holy grail of integrity vs scalability, ensuring a modular, composable, and risk-aware verification that sustains strong integrity for a transaction, which has a cost that is proportional to the number of operations within the transaction, but that does not negatively affect transaction throughput, fault-tolerance, transaction determinism, and auditability.

5 PERFORMANCE EVALUATION AND COMPARATIVE VALIDATION OF THE SCALABLE LOYALTY SETTLEMENT ARCHITECTURE

A multi-regional retail environment with an appropriate structure was intentionally established to represent a range of retail situations, and several of these were simulated and validated for the proposed Scalable Loyalty Settlement Architecture (SLSA). The experimental setup consisted of a cluster of geographically distributed edge nodes and a cluster of settlement shards, which could be configured per application and managed via live synthetic transaction streams that reflected a realistic retail purchasing pattern. It was also responsible for scheduled volume variations, peak-hour surges featuring LTFs, promotions, multi-hit, and 80% of loyalty points delivered, and 20% of multi-party loyalty points picked up and settled. Two possibilities were explored: using a traditional centralized batch reconciliation

engine, which would be used periodically for settlement; and the naive distributed possibility, which would simply replicate the system state across nodes without requiring idempotency or a quorum consensus to commit changes. Besides sustained settlement throughput, the other key performance indicators were the 95th percentile of end-to-end settlement times, reconciliation error rate, mean dispute-resolution time, node scalability (after adding another node), and operational effort (i.e., manual effort per 1 million transactions settled). Experimental results showed that SLSA "performed far better" than the two baselines across all evaluation metrics. Each shard was processed independently to produce its own rewards, rather than by a centralized authority; the deterministic collection of rewards and idempotency eliminated any "double rewards" or "unbalanced rewards". Deterministic settlement was provided by cryptographic ledger proofs, which proved extremely beneficial because they could automatically bring the settlement process to a conclusion, preventing disputes, rather than the manual process. Again, quorum-based settlement is provided, resulting in a distributed, consistent settlement even in the presence of galloping latency, and this is found to be very advantageous, as it can make it easy and effective to bring the settlement process to an end automatically and deterministically, without involving any dispute. There were no integrity controls available for the naive distributed design, and, compared to the centralized processing design, it achieved the highest throughput but also had the highest reconciliation error rate and operational overhead. Distributing a ledger does not mean a secure settlement. The library of experimental tests showcases the objective of a scalable, distributed settlement framework, including necessary deterministic computation, distributed partitioning, cryptographic proof, and input recording to an immutable ledger component, plus advanced speed, accuracy, fault resistance, and operational efficiency in higher-volume retail loyalty applications.

The experimental evaluation is performed on a semi-trained simulation test bed, which provides a consistent setting for a fair comparison between the distributed settlement architecture and the traditional centralized architecture with batch reconciliation, as well as the proposed distributed settlement architecture, named the scalable Loyalty Settlement Architecture (SLSA). The same infrastructure configuration was applied in all the architectures, and the workload was created and deployed synthetically, trying to replicate the real retail customer usage patterns and workloads, which include peak hour activity, promotions, repeat transactions, and a mix of around 80% "accrual" transactions (single party) and around 20% "redemption" transactions (cross party). The one-million-transaction test period applies to all configurations, and numerous tests were performed to obtain reasonably accurate average performance figures. Throughput was measured as the number of transactions successfully committed per second during steady-state operation. Settlement latency was measured as the time required for a transaction to progress from capture and consensus to final ledger commitment. The 95th-percentile latency value was used to evaluate settlement performance under steady-state conditions. A deterministic reference Ledger was created, and the number of transactions that will require corrections for reconciliation when duplicate withholding and inconsistent updates are identified was calculated, along with the percentage. The average time required to resolve intentionally created settlement disputes (mean dispute-resolution time) was used to measure dispute-resolution performance, and scalability was assessed by doubling the number of blocks processed per share and observing the resulting increase in throughput. Operational efficiency was extended to the level of manual intervention required per 1 million transactions, providing us with a complete picture. Various settlement architectures were analyzed here for integrity, which also provided insight into their performance, scalability, and fault tolerance.

Table 2: Comparative Performance Evaluation of Loyalty Settlement Architectures

Performance Metric	Centralized Batch Engine	Naive Distributed Architecture	Proposed SLSA	Observations
Settlement Processing Model	Periodic batch reconciliation	Parallel distributed replication	Deterministic sharded settlement with quorum commitment	SLSA combines distributed execution with integrity verification.
Sustained Throughput (transactions/second)	1,200 tx/s	3,400 tx/s	9,800 tx/s	Independent shard execution enables significantly higher transaction capacity.
Settlement Latency (95th Percentile)	6.4 hours	850 ms	210 ms	Real-time validation removes dependence on delayed batch reconciliation windows.
Reconciliation Error Rate	2.10%	3.45%	0.12%	Idempotency and deterministic accrual nearly

				eliminate duplicate and inconsistent settlements.
Mean Dispute Resolution Time	48 hours	19 hours	1.5 hours	Ledger proofs enable rapid, automated verification rather than manual investigation.
Throughput after Doubling Node Count	1,250 tx/s	5,900 tx/s	18,600 tx/s	SLSA demonstrates near-linear horizontal scalability across distributed shards.
Manual Interventions per 1 Million Transactions	21,000	34,500	1,200	Strong integrity controls substantially reduce operational overhead.
Duplicate Transaction Protection	Limited batch detection	No dedicated mechanism	Full idempotency-key enforcement	Prevents multiple accruals from retries or network failures.
Cross-Shard Consistency	Centralized synchronization	Eventual consistency	Quorum-based deterministic commitment	Eliminates conflicting balance updates across distributed nodes.
Auditability	Log-based manual tracing	Replicated logs	Immutable append-only ledger with Merkle proofs	Enables independent and tamper-evident transaction verification.
Fault Recovery	Batch replay required	Partial state reconstruction	Deterministic replay from ledger evidence	Faster recovery with guaranteed consistency.

Based on the simulation results, the proposed SLSA can effectively address the traditional scalability and transaction integrity challenges in the distributed loyalty settlement system. At a consistency level, one can achieve a good one with an architecture that includes a centralized element, but this might come at the cost of reduced throughput and latency. On the other hand, a naive distributed architecture might achieve high throughput and low latency, but at the cost of not guaranteeing a consistent level. Sliding locks are all that's needed to improve both consistency and performance. In addition, the effectiveness of deterministic partitioning and quorum-based coordination is demonstrated by the fact that horizontal shard expansion provides nearly double the processing power with almost no synchronization overhead. The reduction of reconciliation errors and manual risk interventions is a sign that reconciliation integrity mechanisms are in place to prevent, not solve, reconciliation problems. Moreover, the ledger is immutable, and cryptographic proofs are produced, making dispute resolution easier than ever before, as it is now a verification/confirmation process, rather than an operation-intensive process. Overall, the findings illustrate how an idempotency- and deterministic accrual-based approach, with distributed consensus and tamper-evident ledger architectures, supports efficient processing in very large-scale loyalty ecosystems with a high level of correctness, auditability, and operational resilience.

6 ARCHITECTURAL BENEFITS, SCALABILITY CONSIDERATIONS, AND IMPLEMENTATION TRADE-OFFS OF SLSA

Experimental findings demonstrate that transaction integrity can be integrated directly into the transaction processing lifecycle rather than being treated as a separate post-settlement reconciliation activity. This approach enables integrity verification to occur continuously throughout transaction execution, thereby reducing inconsistencies and improving the reliability of distributed settlement operations. The most significant is that there are considerably fewer rows of cross-debiting and re-crediting stories for reconciliation, which affects operational costs, customer confidence, and accuracy among retail partners operating in a distributed retail environment. All of that has been a form of losing loyalty programs, by using some of the now time-consuming batch reconciliation rules to find duplicate accruals, replayed transactions, and double dings on balances, and then, of course, paying after all the counters have been put in place. The proposed "Scalable Loyalty Settlement Architecture" eliminates those inconsistencies during transaction capture by utilizing digital signatures, idempotency keys, deterministic accrual functions, and, of course, policy validation. So, whereas invalid/duplicate events are just added for consideration for the customer balance, without those time-consuming downstream corrections. The architecture also shows a number of useful throughput enhancements, namely in the domain of processing loyalty increases or accruals. A customer's customer balance would be deterministically

hashed to a specific shard based on the customer token, and most transactions could then be processed concurrently without distributed coordination. Invocations will be used only for transactions that cross organizational/trust boundaries and will cause only a subset of the integrity operations on the ledgers to run (not all). This proportional distribution of integrity costs yields near-linear scalability in the number of processing shards and ensures good architectural correctness. Most notably, these enhancements are at the architectural design level, not in how the technology is deployed, which makes them applicable to other deployments. Furthermore, compared to the naive distributed baseline, the scaling of horizontal augmentation clearly shows that greater complexity and dependence on reconciliation follow from horizontal scaling, thus emphasizing the need for appropriate verifications to build dependable and trustworthy loyalty settlement systems.

Several architectural challenges and trade-offs remain to be addressed. The proposed framework operates within certain constraints that may limit its applicability in some scenarios. Additionally, functional limitations are still being investigated and require further evaluation before large-scale deployment. With distributed systems, the primary compromise is between consistency and availability. This architecture is more about whether they agree and the settlement happens, rather than immediate settlement; i.e., the network can be partitioned, nodes can go offline, so all the nodes must agree before the settlement. Overall, better than settling at the last minute, which may result in a move back, and better than a time-consuming manual effort, which fails to address customer accountability, loyalty, and financial value for the Fintech company. Some groups do promote campaigns where minimal latency matters, but they also don't want to give up the ability to be auditable; in that case, they can select to be more tolerant about requiring consistency on their stores; there, they know that the explicit settlement state machine will provide them with immutable consistency in their low-risk campaigns. The second is the operational complexity. Distributed shards, services that accept consensus, cryptographic services for commitments, immutable ledgers, and continuous infrastructure monitoring demand greater technical capacity and automation expertise than conventional batch systems. In this case, that is, if the business is run like this, any loss of money or disagreements from reconciliation are minimized; however, of course, this has to be done. In addition, the experimental measurements were conducted on a realistic setup with simulated workloads rather than on the actual setup, which could affect the results in a deployment, especially if heterogeneous hardware, volatile networks, non-durable clock synchronization, and/or malicious behavior are present. Lastly, the framework assumes that programs that rely on external resources or non-deterministic decision rules require that those resources be explicitly specified in the program for the program to be verifiable or repeatable. Such restrictions do not detract from the fundamental concept and contribution of the proposed architecture; rather, they serve to guide organizations in realizing system-wide, integrity-maintaining distributed loyalty settlement systems in a scalable way.

7 MANAGERIAL AND TECHNICAL IMPLICATIONS

The proposed Scalable Loyalty Settlement Architecture provides a robust design framework for software architects, platform engineers, and infrastructure teams. It integrates transaction integrity directly into transaction processing rather than relying on end-stage settlement and reconciliation mechanisms. The architecture simplifies downstream settlement and reconciliation processes. It reduces the risk of duplicate transactions, replay attacks, and audit-related inconsistencies in balance records. These improvements are achieved through the use of digital signatures, idempotency keys, and deterministic business logic at the point of transaction capture. Many of these are the same: they make it easier to view bugs and are much more easily automated by using an equal settlement system, regardless of which node the settlement is to be processed on, and are much easier for disaster and settlement system verification. Modular evolution of the system was achieved by splitting each responsibility into ingest, placement, and satisfaction of a satisfactory balance of interests, while keeping a record of transactions on the immutable ledger without performance implications (operating in separate, dedicated engineering teams) [16, 17]. The architecture supports incremental scalability through the addition of processing shards without requiring major infrastructure redesign. This approach minimizes deployment risks and enables seamless capacity expansion. Furthermore, reconciliation and dispute resolution rely on cryptographic proof verification rather than labor-intensive manual auditing, improving operational efficiency and reliability. Effective partition-key design is essential for achieving efficient distributed processing. Operations should be localized to the minimum number of shards possible, thereby reducing cross-shard coordination and improving overall system performance and scalability. For engineering teams, it's important that they keep themselves informed of the speed of the fast-path execution ratio, frequency of participation by the quorum, and the speed of the quorum's reconciliation and the number of shards, so they can enhance their performance without compromising the integrity guarantees and long-term goals of their design.

From a strategic and managerial perspective, this architecture will be the real-time digital capability that helps build customer trust, create stronger partnerships, and encourage sustainable business growth. Most of the cost of loyalty platforms stems from several common issues, such as miscalculating customer reward balances (which cause customer service expenses), willful settlement delays, manual settlement reconciliation, and dispute handling. The proposed framework tends to reduce the risk of reconciliation errors, manual processing, or the resolution of discrepancies,

improve customer experience and quality, and boost the reliability of the customer engagement strategy in their loyalty program [18, 19]. It brings a cryptographically guaranteed real-time settlement, which allows the retailer, financial institution, and marketplace operator to build massive coalition loyalty ecosystems that easily convert loyalty from a brand to a marketplace partner without the need for delayed batch reconciliation or reliance on a central assumption. This functionality makes it easier to implement new business formats and, for example, to define cross-brand reward programs/exchanges, multi-retailer promotions, flows to partner markets, and regional expansion programs that could not have been implemented if there had been a need to forgo high settlement risk and high administrative overload. To achieve the long-term benefits of operation's Scalability, Regulatory Compliance, audit-ready, Financial Risk Management, and Agility, a set of consensus modules, distributed shards, and immutable ledgers, along with the constant monitoring infrastructure, will be viewed as an increase in organizational maturity while making an investment. However in the implementation of settlement architectures alongside executives and technology decision makers, it is important to try and consider this not simply one technology upgrade (or enhancement) but think about it as being one element in the digital infrastructure that will need to offer the trust, scale and interactivity that will be required towards the long-term development of the loyalty programs, partnerships and customer-driven innovation, in an increasingly dispersed retail landscape.

8 FUTURE RESEARCH DIRECTIONS

As retail commerce, cloud computing, and intelligent automation continue to grow and mature rapidly, there is really no reason anyone could not use the proposed Scalable Loyalty Settlement Architecture (SLSA) to meet the integrity and scalability requirements for loyalty settlement in a distributed retail environment. A new possibility lies in restructuring and self-organization within the settlement life cycle, with an artificial intelligence element. These models of machine learning prediction can predict transactions in the future that are passed on to other businesses, maybe even predict the number of promotions (with a fixed set of business partners), and prepare all transactions in advance so that, even if we have to handle an extremely dynamic load, we have much less variation in an income stream and very consistent income. The proposed approach continuously monitors historical shard allocation patterns and dynamically adjusts shard assignments based on real-time network conditions [20]. By leveraging insights from past network behavior and operational performance, it optimizes shard allocation, quorum selection, routing decisions, and overall network efficiency beyond what traditional static assignment algorithms can achieve. The other big opportunity lies in implementing methods that enable the 'privacy preserving computation' of multiple retailers and coalition partners to jointly, privately, yet without knowing anything valuable about the business, validate a loyalty transaction. Another would be to explore alternatives to consensus mechanisms that can configure quorum types based on transaction values, the trusted status of organizations, or even network availability, aiming to achieve greater performance and resilience. Furthermore, if verifiable credentials, blockchain-based cryptographic attestation mechanisms, and programmable tokens are integrated into interfaces/connectors with decentralized identity systems, interoperability can be achieved without compromising or losing guarantees of deterministic settlement. All these additions would also include Intelligent Anomaly Detection, Automated Fraud Prevention, and a Self-Verifying Transaction Pipeline, which would not only make them robust but also secure. The addition of all these would be a new suite of features that would make it more secure to use, notifying the user as he/she enters into the settlement process. These advances offer opportunities for transactional integrity in millions of transactions across the globe, transactions that may occur in "retail store" miles apart - and for intelligent and autonomous platforms that can support the fast-changing and varied channels of the retail industry and the loyalty journey that traverses them. Future work may also include studying quantum-resistant cryptographic primitives and digital signatures that withstand quantum attacks to ensure long-term security amid the evolving paradigm of information processing and the emergence of new security risks, while equipping loyalty programs to build the next generation of secure digital transactions in the realm of traditional cryptography.

Still another future roadmap would be the knowledge-based approach (international standards on interoperability), in which knowledge would be gained about the evolution of interoperability processes beyond retail and traditional approaches to the use of interoperability standards (IS). Unified value-sharing networks are increasingly connecting retailers, financial institutions, healthcare providers, telecom operators, e-commerce platforms, and government-backed reward programs. In these ecosystems, rewards, vouchers, and loyalty points serve as common value-exchange mechanisms, enabling seamless interactions across multiple sectors. These ecosystems require interoperable frameworks that support data sharing across multiple ledgers and standardized communication protocols. They should incorporate programmable settlement contracts and policy-aware orchestration mechanisms to enable secure, consistent, and verifiable value exchange across participating entities. It is important to find a common denominator among the settlements, including further research into the potential of a common currency, thereby enabling easy transfers of loyalty tokens that aren't managed by a trusted third party. All this is important, but scaling and operating in non-standard H/W, EDGE (near the edge of connectedness), hostile environments, and intermittent connectivity (time-varying connectivity) (longer durations) is important, too. In addition, researchers can analyze how to build architectures that allow a node to fail and then partition into two nodes, re-reset the cryptographic commitments, re-

split the load, and deterministically (not randomly) recover the two nodes without human intervention. Finally, the development of energy-aware shard allocation, carbon-efficient consensus protocol design, and flexible resource programming to reduce infrastructure costs while maintaining high settlement performance is significantly extended and connected to the concepts of “sustainable smart city”. Moreover, if an energy-efficient consensus protocol (energy-efficient reduction tagging of infrastructure costs) and energy-aware shard allocation are applied, energy efficiency and carbon-efficient reduction of the infrastructure costs are also tackled as a result of both of those approaches, which are called the concept of sustainable smart city (not dealing with green computing) and flexible resource programming, which reduce the costs of the infrastructure. Not only that, but the power of EAI can also be used in conjunction with transaction monitoring and dispute resolution, as well as compliance reporting, with the ability to provide us with an explanation of why the automated settlement took place, transparent to both regulators, auditors, business partners, and customers. A Digital Value Network (DVN) can integrate IoT-enabled retail devices, digital wallets, central bank digital currencies (CBDCs), and smart contracts into a unified settlement ecosystem. Such an approach can reduce reconciliation costs, improve the accuracy of reward and payout calculations, and provide a secure and transparent platform for managing omnichannel retail operations, coalition loyalty programs, and cross-border reward networks.

9 CONCLUSION

To address the challenge of enterprise applications in the vast, distributed retail landscape, where maintaining transaction integrity is increasingly difficult, the authors of this paper proposed a new Scalable Loyalty Settlement Architecture (SLSA). The architecture incorporates tokenized identity management, idempotent transaction processing, and deterministic loyalty value accrual. It utilizes quorum-based settlement orchestration and cryptographic Merkle proofs to maintain ledger integrity. These mechanisms enable secure value creation, transfer, redemption, and verification throughout the transaction lifecycle. The accompanying Transaction Integrity Framework also ensures the system's dependability by establishing stages of proof generation interlaced with a fixed state machine, enabling predictable recovery, efficient conflict resolution, and resilience to component failures and system issues. The study demonstrated substantial performance improvements over centralized batch-processing and conventional distributed approaches. Deterministic sharding enabled near-linear horizontal scalability, higher throughput, and reduced reconciliation errors. Because these improvements are achieved at the architectural level, the framework can be adapted to diverse retail environments. Despite considerations such as operational complexity, deterministic rule requirements, and quorum-based consistency, integrity-first settlement architectures are both technically feasible and economically advantageous. As a result, they provide a foundation for secure, transparent, scalable, and trustworthy value exchange across modern retail and multi-partner digital commerce ecosystems.

STATEMENTS ON COMPLIANCE WITH ETHICAL STANDARDS

Disclosure of conflict of interest

No conflict of interest to be disclosed.

REFERENCES

- [1] Van Steen, M. (2002). Distributed systems: Principles and paradigms. Network, 2(28), 1.
- [2] Bernstein, P. A., & Newcomer, E. (2009). Principles of transaction processing. Morgan Kaufmann.
- [3] Lamport, L. (2019). The part-time parliament. In Concurrency: The Works of Leslie Lamport (pp. 277–317). ACM.
- [4] Helland, P. (2012). Idempotence is not a medical condition. Communications of the ACM, 55(5), 56–65.
- [5] Corbett, J. C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J. J., et al. (2013). Spanner: Google's globally distributed database. ACM Transactions on Computer Systems, 31(3), 1–22.
- [6] Gray, J., & Reuter, A. (1992). Transaction processing: Concepts and techniques. Elsevier.
- [7] Kleppmann, M. (2017). Designing data-intensive applications: The big ideas behind reliable, scalable, and maintainable systems. O'Reilly Media.
- [8] Vogels, W. (2009). Eventually consistent. Communications of the ACM, 52(1), 40–44.
- [9] Stopford, B. (2018). Designing event-driven systems. O'Reilly Media.
- [10] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.
- [11] Brewer, E. (2012). CAP twelve years later: How the “rules” have changed. Computer, 45(2), 23–29.

- [12] Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. In Proceedings of the 2014 USENIX Annual Technical Conference (USENIX ATC 2014) (pp. 305–319).
- [13] Merkle, R. C. (1987). A digital signature based on a conventional encryption function. In Advances in Cryptology – EUROCRYPT '87 (pp. 369–378). Springer, Berlin, Heidelberg.
- [14] Breugelmans, E., Bijmolt, T. H., Zhang, J., Basso, L. J., Dorotic, M., Kopalle, P., et al. (2015). Advancing research on loyalty programs: A future research agenda. *Marketing Letters*, 26(2), 127–139.
- [15] Cai, W., Wang, Z., Ernst, J. B., Hong, Z., Feng, C., & Leung, V. C. (2018). Decentralized applications: The blockchain-empowered software system. *IEEE Access*, 6, 53019–53033.
- [16] Fowler, M. (2012). Patterns of enterprise application architecture. Addison-Wesley.
- [17] Newman, S. (2021). Building microservices: Designing fine-grained systems (2nd ed.). O'Reilly Media.
- [18] Castro, M., & Liskov, B. (1999). Practical Byzantine fault tolerance. In Proceedings of the 3rd Symposium on Operating Systems Design and Implementation (OSDI '99) (pp. 173–186).
- [19] Dorotic, M., Bijmolt, T. H., & Verhoef, P. C. (2012). Loyalty programmes: Current knowledge and research directions. *International Journal of Management Reviews*, 14(3), 217–237.
- [20] Kumar, V., & Reinartz, W. (2018). Customer relationship management: Concept, strategy, and tools (3rd ed.). Springer.