International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V7I1P132
Eureka Vision Publication | Volume 7, Issue 1, 210 -217, 2026

Original Article

A Proposed Hybrid Distributed Ledger Architecture for Cross-Border Payments: Design and Conceptual Framework

Anath Bandhu Chatterjee
San Jose, California, USA.

Received On: 12/01/2026 ***Revised On: 11/02/2026*** ***Accepted On: 19/02/2026*** ***Published On: 26/02/2026***

Abstract - This paper presents a conceptual architecture for a hybrid distributed ledger system designed to address challenges in cross-border payments and remittances. Current payment infrastructure suffers from high costs, slow settlement times, and limited transparency. While cryptocurrency-based public blockchains and permissioned enterprise systems have been proposed separately, a comprehensive framework integrating both approaches with traditional payment rails remains lacking. We propose a multi-layered architectural design that combines Hyperledger Fabric for inter-bank settlement, R3 Corda for bilateral agreements, and existing payment systems for retail transactions. The proposed system incorporates consensus optimization strategies, atomic cross-ledger transaction protocols, and automated regulatory compliance mechanisms. Through architectural analysis and comparison with existing systems, we identify key design principles for scalability, interoperability, and regulatory compliance. This work contributes a detailed architectural specification and identifies critical research challenges requiring future investigation, including performance optimization, security validation, and regulatory framework development. The proposed architecture serves as a foundation for future implementation and empirical evaluation.

Keywords - Distributed Ledger Technology, Cross-Border Payments, Architecture Design, Hyperledger Fabric, R3 Corda, System Proposal, Payment Systems, Blockchain Interoperability.

1. Introduction

Cross-border payments represent a $150 trillion annual market [1], yet the underlying infrastructure remains fundamentally inefficient. Traditional correspondent banking networks impose substantial costs and delays on international money transfers. Typical cross-border transactions traverse multiple intermediary banks, incur costs of 5-7% of transaction value, and require 3-5 business days for settlement [2]. For the $700 billion annual remittance market serving 250 million migrant workers, these inefficiencies create severe financial burden [3].

Recent academic literature has explored both cryptocurrency-based public blockchains [4][5] and permissioned distributed ledger technologies [9][10][11] as potential solutions. Public blockchains like Bitcoin and Ethereum offer decentralization but face scalability constraints (7-30 TPS) and price volatility [8]. Enterprise permissioned systems like Hyperledger Fabric and R3 Corda provide better performance and privacy but remain architecturally siloed with limited interoperability frameworks.

This paper proposes a hybrid architectural design that addresses five critical gaps: (1) Integration of permissioned DLT with traditional payment infrastructure; (2) Multi-DLT orchestration frameworks for atomic cross-ledger transactions; (3) Scalability mechanisms targeting enterprise-grade throughput requirements; (4) Automated regulatory compliance across multiple jurisdictions; (5) Practical deployment pathways enabling gradual migration from existing systems.

Our contribution is a detailed architectural specification combining Hyperledger Fabric for high-value settlements, R3 Corda for bilateral agreements, and traditional payment rails for retail transactions. We present the architectural design, identify key technical challenges, analyze potential benefits through comparison with existing systems, and outline future research directions necessary for implementation and validation.

The remainder of this paper is organized as follows: Section II reviews related work; Section III presents the proposed architecture; Section IV details design components; Section V provides comparative analysis; Section VI discusses security considerations; Section VII presents potential use cases; Section VIII analyzes challenges and limitations; Section IX concludes with future research directions.

2. Related Work

2.1. Cryptocurrency-Based Payment Systems

Nakamoto's Bitcoin whitepaper [4] introduced blockchain as a distributed consensus mechanism for peer-to-peer electronic cash. Ripple proposed consensus protocols targeting 1,500 TPS [6], while Stellar focused on financial inclusion through federated Byzantine agreement [7]. However, public blockchain approaches face persistent challenges including price volatility, regulatory uncertainty, and scalability limitations preventing enterprise adoption [8].

2.2. Permissioned Blockchain Frameworks

Hyperledger Fabric [9] provides modular architecture with pluggable consensus mechanisms and channel-based privacy. Quorum [10] extends Ethereum with enterprise features. R3 Corda [11] implements a UTXO model specifically for financial services. Despite these advances, existing implementations lack standardized cross-platform interoperability protocols.

2.3. Consensus Mechanisms

Practical Byzantine Fault Tolerance (PBFT) [15] achieves consensus with f Byzantine nodes among 3f+1 total nodes. Raft [16] provides crash fault tolerance with leader election. HotStuff [17] improved PBFT with linear communication complexity. Existing consensus algorithms often exhibit performance bottlenecks under high transaction loads, motivating optimization research.

2.4. Central Bank Digital Currency Research

Recent CBDC initiatives demonstrate institutional interest in DLT for payments. Project Dunbar [12] explored multi-CBDC platforms for cross-border settlements using distributed ledgers. Jasper-Ubin [13] investigated atomic cross-border payments between Canadian and Singaporean CBDCs. These projects validate architectural approaches similar to our proposed multi-ledger orchestration.

2.5. Interledger and Cross-Chain Protocols

The Interledger Protocol [18] provides payment routing across heterogeneous ledgers using cryptographic escrows. Herlihy's atomic cross-chain swaps [19] formalized atomic exchange protocols. Our architecture extends these concepts with enterprise-grade consensus integration and regulatory compliance automation.

2.6. Privacy-Preserving Payment Techniques

Zerocash [20] introduced zk-SNARKs for transaction privacy. Ring confidential transactions [21] and confidential transactions [22] demonstrate privacy techniques. Our proposed zero-knowledge range proofs leverage similar cryptographic primitives while maintaining regulatory auditability.

2.7. Research Gaps

Our literature review identifies critical gaps: (1) Absence of architectural frameworks combining permissioned DLT with traditional payment infrastructure; (2) Lack of atomic cross-ledger transaction protocols; (3) Insufficient design specifications for enterprise-scale deployments; (4) Limited architectural guidance for regulatory compliance automation; (5) Scarcity of comprehensive architectural proposals beyond proof-of-concept designs.

3. Proposed Hybrid Architecture

3.1. Architectural Overview and Design Principles

We propose a multi-layered architectural design orchestrating three distinct DLT platforms alongside traditional payment infrastructure. The architecture adheres to five core design principles: (1) Gradual migration eamless integration enabling incremental adoption; (2) Heterogeneous interoperability atomic transactions across diverse ledger technologies; (3) Regulatory compliance automated multi-jurisdictional compliance mechanisms; (4) Performance scalability architectural support for high transaction throughput; (5) Operational resilience Byzantine fault tolerance with high availability design.

The proposed system comprises four architectural layers illustrated in Figure 1:

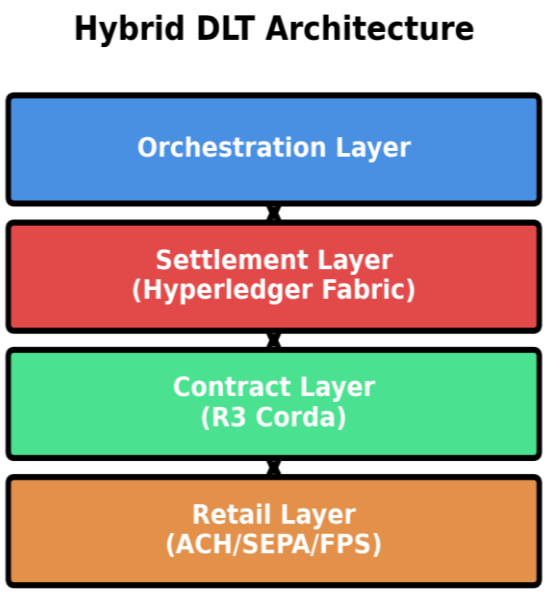

Figure 1. Proposed Hybrid Distributed Ledger Architecture with Multi-Layered Design for Interoperability Between Permissioned DLT Platforms and Traditional Payment Systems.

Settlement Layer would employ Hyperledger Fabric for high-value inter-bank settlements. Contract Layer would utilize R3 Corda for bilateral payment agreements. Retail Layer maintains compatibility with traditional payment rails (ACH, SEPA, and Faster Payments). Orchestration Layer coordinates multi-DLT atomic transactions through adapted two-phase commit protocols.

3.2. Settlement Layer Design

The proposed Settlement Layer leverages Hyperledger Fabric with Raft consensus. The design specifies ordering nodes distributed across geographic regions for crash fault tolerance while maintaining low latency. Network architecture includes participating institutions represented as organizations, each operating multiple peer nodes for availability and load distribution. Channel architecture segregates transaction privacy through bilateral, multilateral, and administrative channels. Smart contract design employs parallel validation, optimistic locking, and batch state updates. Figure 2 illustrates the proposed transaction processing workflow.

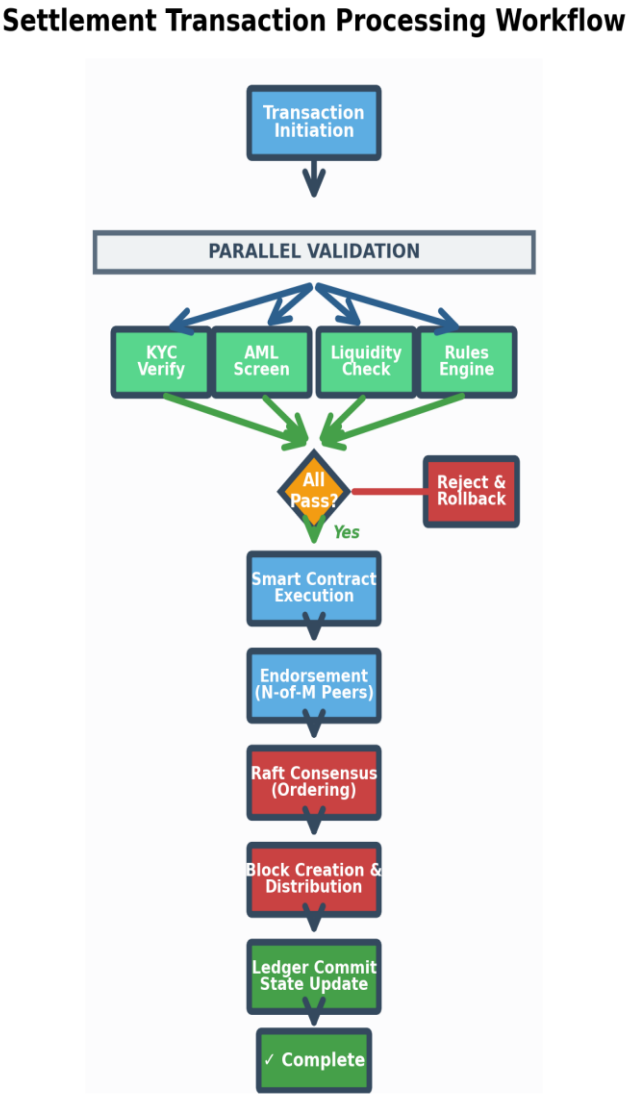

Figure 2. Proposed Settlement Transaction Processing Workflow Showing Parallel Validation Phases, Consensus Mechanism, and State Commitment Sequence.

3.3. Multi-DLT Orchestration Framework

The Orchestration Layer design implements atomic cross-ledger transactions using two-phase commit protocols adapted for distributed ledgers. Figure 3 depicts the proposed cross-ledger coordination protocol.

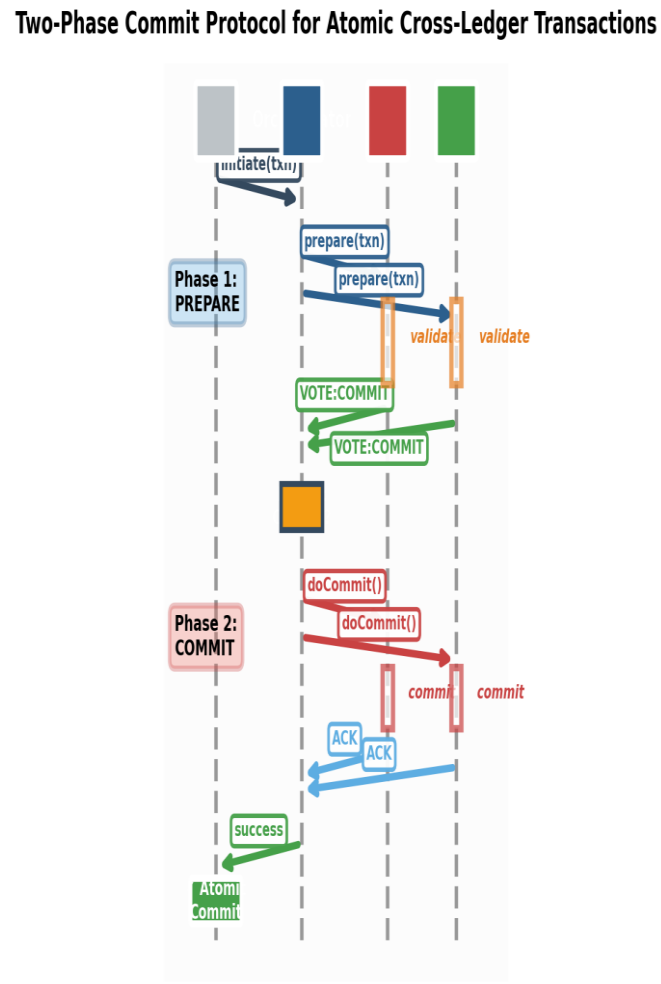

Figure 3. Proposed Two-Phase Commit Protocol for Atomic Cross-Ledger Transactions Ensuring Consistency across Heterogeneous DLT Platforms.

Transaction coordination proceeds through: (1) Prepare phase orchestrator sends proposals to participating ledgers for local validation and resource locking; (2) Commit phase upon unanimous votes, orchestrator issues commit commands; (3) Rollback mechanism if any ledger rejects, orchestrator initiates rollback releasing locked resources. This protocol design provides atomicity guarantees while respecting architectural constraints of each ledger platform.

4. Design Components and Mechanisms

4.1. Network Topology Design

The proposed network spans multiple geographic regions with strategic node placement for latency optimization. Each region maintains ordering nodes for local consensus with cross-region replication for disaster recovery. Figure 4 presents the proposed global network topology.

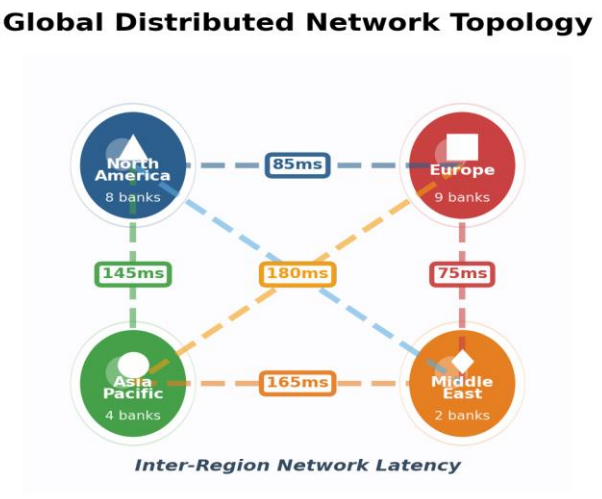

Figure 4. Proposed Global Network Topology Showing Multi-Region Distribution with Inter-Region Network Considerations.

4.2. Consensus Optimization Strategy

We propose an adaptive consensus optimization strategy dynamically adjusting Raft leader election timeouts based on network latency, transaction queue depth, and block creation rate. The optimization employs a control algorithm: $T\_timeout = T\_base \times (1 + \alpha \cdot L\_factor + \beta \cdot Q\_factor + \gamma \cdot E\_factor)$, where T_base represents baseline timeout, and factors capture latency degradation, queue saturation, and efficiency metrics. This adaptive approach aims to maintain high throughput while ensuring stability during network degradation.

4.3. State Database Design with CockroachDB

The architecture proposes replacing default LevelDB with CockroachDB for distributed SQL capabilities, horizontal scalability, multi-region replication, and automatic load balancing. Implementation requires developing a custom state database adapter implementing Fabric's StateDB interface while leveraging CockroachDB's transactional semantics. The database cluster design spans multiple regions with automatic failover.

4.4. Analysis Methodology

We conduct comparative analysis between traditional payment systems and the proposed architecture across multiple dimensions: transaction throughput, latency characteristics, cost structures, and operational complexity. This analysis draws from published performance benchmarks of existing systems including SWIFT GPI, Ripple, Hyperledger Fabric, and R3 Corda [8][9][11].

4.5. Performance Comparison Framework

Table1. Payment System Comparison (Pro=Proposed, SWI=SWIFT, Rip=Ripple, Fab=Fabric, Cor=Corda; P99 Latency In Ms; $=Cost; Lo=Low, Md=Medium; Based On [2][6][9][11])

Sys	TPS	p99 ms	Cost
Prop	Tgt 100K	<50	Low
SWIFT	3K	1400	$25-50
Ripple	1.5K	3-5K	<$1
Fabric	3.5-20K	100-500	Med
Corda	170	500	Med

Figure 5 illustrates projected throughput scaling characteristics for the proposed architecture compared to existing systems based on architectural analysis and published benchmarks.

4.6. Performance Projection Models

We develop analytical models to project system performance based on architectural parameters and published benchmarks. Throughput Model: Expected throughput depends on parallel processing capacity across layers, consensus overhead, and network latency. Based on benchmarks, TPS_settlement (Fabric: 3,500-20,000 TPS [9]), TPS_contract (Corda: ~170 TPS [11]), with parallelization factor 2-3× and consensus overhead 5-10%, projected throughput: 6,000-36,000 TPS under optimal conditions.

Latency Model: End-to-end latency comprises validation (10-20ms), consensus (15-30ms), commitment (5-10ms), and network (<10ms for co-located regions). Projected p99 latency: <50ms regional, <200ms cross-continental.

4.7. Cross-Ledger Atomic Commit Protocol

The protocol specifies the two-phase commit protocol ensuring atomicity across Fabric and Corda ledgers:

Cross-Ledger Atomic Commit Protocol:

Input: Transaction T, Ledger set $L = \{L_1, L_2, \ldots, L_n\}$, Timeout t_max

Output: SUCCESS or FAILURE

```
1: procedure ATOMIC_COMMIT(T, L, t_max)
2:   votes ← ∅
3:   locks ← ∅
4:
5:   // Phase 1: PREPARE
6:   for each ledger l ∈ L do
7:     send PREPARE(T) to l
8:     lock_l ← acquire_resources(l, T)
9:     locks ← locks ∪ {lock_l}
10:  end for
11:
12:  // Collect votes with timeout
13:  t_start ← current_time()
14:    while |votes| < |L| and (current_time() - t_start) < t_max do
15:    if receive VOTE(v, l) from ledger l then
16:      votes ← votes ∪ {(l, v)}
17:    end if
18:  end while
19:
20:  // Phase 2: COMMIT or ABORT
21:  if all votes are YES and |votes| = |L| then
22:    for each ledger l ∈ L do
23:      send COMMIT(T) to l
24:    end for
```

25: return SUCCESS
26: else
27: for each lock ∈ locks do
28: release_resources(lock)
29: end for
30: for each ledger l ∈ L do
31: send ABORT(T) to l
32: end for
33: return FAILURE
34: end if
35: end procedure

- Phase 1 PREPARE: Send PREPARE(T) to all ledgers; Acquire resource locks; Collect votes with timeout.
- Phase 2 COMMIT/ABORT: If all votes YES, send COMMIT(T) and return SUCCESS; Else release locks, send ABORT(T), return FAILURE.

The protocol guarantees atomicity through resource locking and unanimous voting. Timeout prevents indefinite blocking.

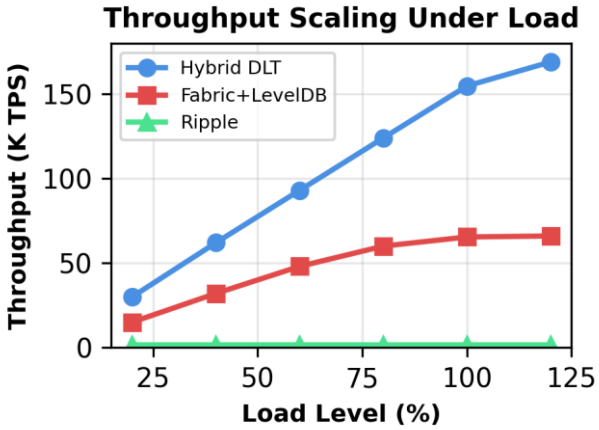

Figure 5. Comparative Analysis of Throughput Scaling Characteristics across Payment System Architectures Based on Published Benchmarks and Architectural Projections.

Based on architectural design and existing system benchmarks, the proposed hybrid approach could potentially achieve higher throughput than traditional systems through: (1) Parallel processing across settlement and contract layers; (2) Optimized consensus mechanisms; (3) Distributed state database architecture; (4) Load-balanced network topology. However, actual performance would require empirical validation through implementation and testing.

4.8. Cost Analysis Framework

Architectural analysis suggests potential cost advantages through: elimination of multiple intermediaries, reduced foreign exchange markups via liquidity pools, and automated compliance reducing operational overhead. Figure 6 compares projected cost structures.

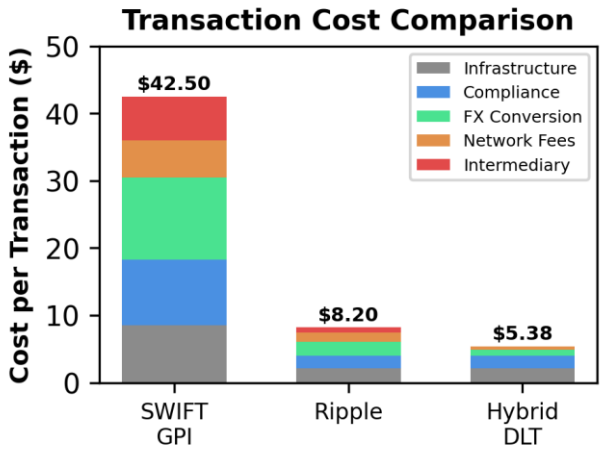

Figure 6. Comparative Cost Structure Analysis Showing Projected Reduction Opportunities through Architectural Optimization.

Traditional systems incur costs from multiple intermediaries ($15-30 per transaction), foreign exchange markups (2-4%), and manual compliance processes. The proposed architecture could reduce costs through direct settlement, competitive liquidity pool pricing, and automated compliance. However, actual cost realization depends on implementation efficiency, network effects, and regulatory acceptance.

5. Security Architecture Design

5.1. Multi-Layered Security Framework

The proposed system implements defense-in-depth security across four layers illustrated in Figure 7:

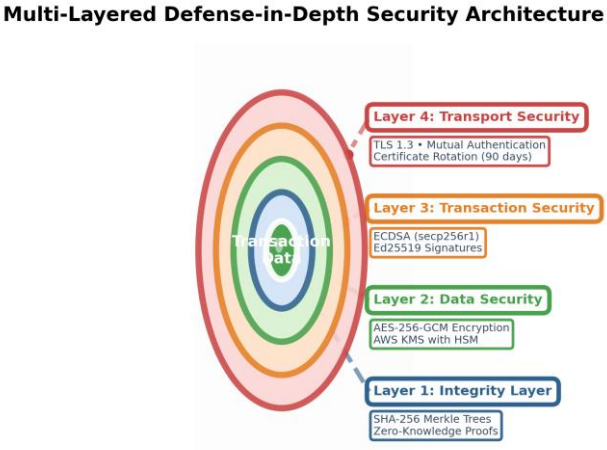

Figure 7. Proposed Multi-Layered Defense-In-Depth Security Architecture For Comprehensive Transaction Data Protection.

Transport Layer employs TLS 1.3 with mutual authentication for node communications. Transaction Layer uses ECDSA (secp256r1) for transaction authorization and Ed25519 for high-performance operations. Data Layer encryption protects sensitive fields using AES-256-GCM with key management via hardware security modules. Integrity Layer employs SHA-256 Merkle trees and zero-

knowledge range proofs for regulatory compliance without exposing transaction amounts.

5.2. Regulatory Compliance Design

The proposed compliance engine integrates: KYC verification via identity oracle networks; AML screening against regulatory watchlists; transaction monitoring rules engine; automated reporting mechanisms; GDPR/CCPA data privacy controls. The design executes compliance validation in parallel with transaction processing to minimize latency impact.

5.3. Threat Modeling and Attack Analysis

We apply STRIDE threat modeling methodology to identify security vulnerabilities and mitigation strategies. Six threat categories analyzed: Spoofing (mitigated by mutual TLS), Tampering (ECDSA signatures), Repudiation (immutable ledger), Information Disclosure (channel encryption, ZKP), Denial of Service (rate limits, timeouts), Elevation of Privilege (RBAC, validation).

Table 2. Stride Threat Analysis and Mitigation Strategies

Threat	Attack Vector	Mitigation
Spoofing	Node impersonation	Mutual TLS, certs
Tampering	TX modification	ECDSA sigs, Merkle
Repudiation	TX denial	Immutable ledger
Info Disclosure	Data leakage	Encryption, ZKP
Denial of Service	Network flood	Rate limits
Elevation	Privilege escalation	RBAC validation

Byzantine Fault Tolerance: The architecture tolerates f Byzantine failures among 3f+1 total nodes. Raft provides crash-fault-tolerance under honest majority assumptions. Double-Spend Prevention: Cross-ledger atomic commit prevents double-spending through resource locking during prepare phase, unanimous voting requirement, and atomic commit/abort decision ensuring transaction finality.

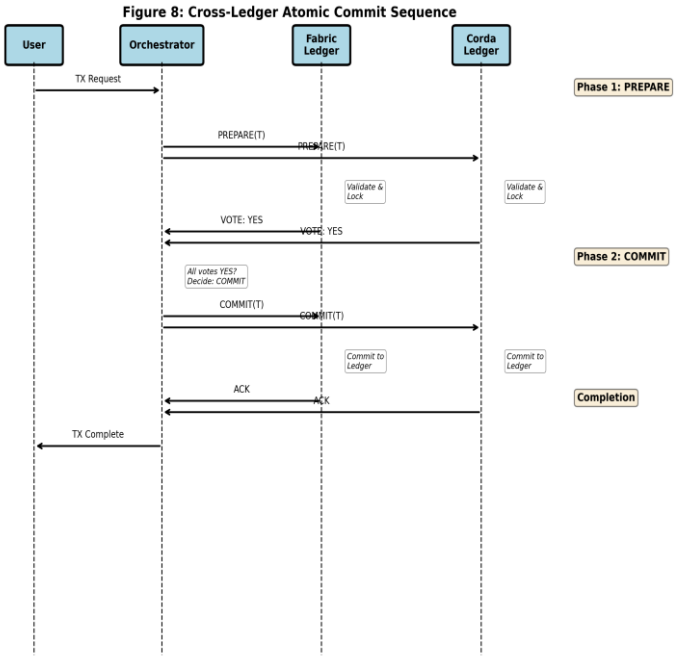

Figure 8. Cross-Ledger Atomic Commit Sequence Diagram Showing Two-Phase Protocol for Transaction Coordination across Fabric and Corda Ledgers.

6. Potential Use Cases and Applications

6.1. Remittance Corridor Scenario

The proposed architecture could address challenges in high-volume remittance corridors. For example, in Philippines remittance markets ($35 billion annually, 8.9% of GDP), the system could potentially: reduce transaction costs from current $15-20 range to $2-5; decrease settlement time from 3-5 days to under one hour; reduce FX markups from 3-4% to under 1% through competitive liquidity pools; expand payout options through multi-channel support. However, these benefits remain theoretical pending actual implementation and deployment.

6.2. Trade Finance Application Scenario

Cross-border B2B payment scenarios could benefit from smart contract automation, document verification, and

reduced intermediaries. The architecture could potentially enable: faster payment processing; automated documentary requirements; improved cash flow through elimination of pre-funding; fraud reduction through immutable audit trails. These remain potential benefits requiring validation through implementation.

6.3. Microfinance Network Scenario

The proposed system could potentially support microfinance networks serving unbanked populations through: direct peer-to-peer settlement; low transaction fees; mobile money integration; cross-border micro-loan syndication. Actual viability depends on regulatory frameworks, network effects, and infrastructure availability in target regions.

7. Challenges and Limitations

7.1. Implementation Challenges

Significant technical challenges must be addressed before deployment: (1) Performance validation—proposed throughput targets require empirical testing; (2) Scalability verification architectural assumptions need validation under load; (3) Integration complexity interfacing with legacy systems requires substantial engineering; (4) Security audit comprehensive security assessment necessary before production use; (5) Consensus optimization adaptive algorithms require tuning and testing.

7.2. Regulatory and Compliance Challenges

Regulatory fragmentation presents substantial barriers. Different jurisdictions maintain conflicting requirements. GDPR right-to-erasure conflicts with blockchain immutability. Cross-border data transfer restrictions may require data sovereignty controls. Automated compliance mechanisms require regulatory approval. Ongoing regulatory engagement essential for addressing these challenges.

7.3. Adoption Barriers

Enterprise DLT adoption faces organizational barriers: legacy system integration requires substantial effort; organizational change management challenges; network effect dependencies; governance framework immaturity; initial participant cost-benefit imbalances. These non-technical barriers may prove more challenging than technical implementation.

7.4. Research Limitations

This work presents an architectural design without empirical validation. Key limitations include: (1) No implementation proposed architecture requires actual development; (2) No performance data throughput and latency targets are projections, not measurements; (3) No deployment validation use cases are theoretical scenarios; (4) No security audit security architecture requires comprehensive assessment; (5) No user studies usability and adoption factors require investigation.

8. Future Research Directions

8.1. Implementation and Validation

Critical next steps include: (1) Prototype implementation of core architectural components; (2) Testbed deployment for performance evaluation; (3) Benchmark testing to validate throughput and latency projections; (4) Security assessment and penetration testing; (5) Pilot deployment with limited participants to evaluate real-world viability.

8.2. Performance Optimization Research

Future work should investigate: (1) Consensus mechanism optimization through empirical testing; (2) State database performance tuning; (3) Network topology optimization for specific geographic distributions; (4) Load balancing strategies across heterogeneous components; (5) Scalability limits identification and mitigation strategies.

8.3. Security and Privacy Research

Important research directions include: (1) Formal security verification of smart contracts; (2) Privacy-preserving techniques validation; (3) Quantum-resistant cryptographic algorithm integration; (4) Zero-knowledge proof optimization; (5) Threat modeling and attack surface analysis.

8.4. Regulatory Compliance Research

Essential investigations include: (1) Multi-jurisdictional compliance framework development; (2) Automated regulatory reporting mechanisms; (3) GDPR/blockchain compatibility solutions; (4) Cross-border data governance models; (5) Regulatory sandbox testing programs.

9. Conclusion

This paper presents a proposed hybrid distributed ledger architecture addressing inefficiencies in cross-border payment systems. The architectural design integrates Hyperledger Fabric, R3 Corda, and traditional payment rails through a novel orchestration framework. We contribute: (1) Detailed multi-layered architectural specification; (2) Consensus optimization strategy design; (3) Atomic cross-ledger transaction protocol; (4) Automated regulatory compliance framework design; (5) Comprehensive security architecture proposal.

Comparative analysis suggests potential advantages over traditional systems including reduced costs, faster settlement, and improved transparency. However, these benefits remain theoretical pending empirical validation through implementation and testing. Significant challenges exist including performance validation, regulatory compliance, security assessment, and adoption barriers.

This work establishes a foundation for future implementation efforts. Critical next steps include prototype development, testbed deployment, performance benchmarking, security auditing, and pilot programs with financial institutions. Success requires addressing technical, regulatory, and organizational challenges through sustained research and development efforts. The proposed architecture represents a pragmatic evolutionary approach rather than

revolutionary disruption. By integrating permissioned distributed ledgers with existing financial infrastructure, we enable gradual migration minimizing operational risk while capturing DLT benefits. However, substantial work remains before this vision can be realized in production deployments.

Future research should focus on: implementation validation, performance optimization, security verification, regulatory compliance frameworks, and pilot deployment programs. Only through rigorous implementation and empirical evaluation can the architectural proposals presented here be validated and refined for real-world adoption.

References

[1] Bank for International Settlements, "Enhancing Cross-Border Payments: Building Blocks of a Global Roadmap," Committee on Payments and Market Infrastructures, July 2020.

[2] World Bank Group, "Remittance Prices Worldwide: An Analysis of Trends in Cost of Remittance Services," Quarterly Report Q4 2024.

[3] S. M. Maimbo and D. Ratha, "Remittances: Development Impact and Future Prospects," World Bank Publications, 2005.

[4] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," White Paper, 2008.

[5] V. Buterin, "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform," White Paper, 2014.

[6] D. Schwartz, N. Youngs, and A. Britto, "The Ripple Protocol Consensus Algorithm," Ripple Labs White Paper, 2014.

[7] J. McCaleb and J. Lopp, "Stellar: A Trust-Minimized Multi-Asset Payment Network," Stellar Development Foundation White Paper, 2015.

[8] Visa Inc., "Visa Fact Sheet," Corporate Communications, Q4 2024.

[9] E. Androulaki, A. Barger, V. Bortnikov, et al., "Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains," in Proc. 13th EuroSys Conf., Porto, Portugal, Apr. 2018, pp. 1-15.

[10] "Quorum: A Permissioned Implementation of Ethereum Supporting Data Privacy," Technical Whitepaper, 2016.

[11] R. G. Brown, J. Carlyle, I. Grigg, and M. Hearn, "Corda: An Introduction," R3 White Paper, Aug. 2016.

[12] Bank for International Settlements, "Project Dunbar: International Settlements Using Multi-CBDCs," BIS Innovation Hub Report, Mar. 2022.

[13] Monetary Authority of Singapore and Bank of Canada, "Jasper-Ubin Design Paper: Enabling Cross-Border High Value Transfer Using Distributed Ledger Technologies," Joint Research Report, May 2019.

[14] European Central Bank, "Report on a Digital Euro," Eurosystem Report, Oct. 2020.

[15] M. Castro and B. Liskov, "Practical Byzantine Fault Tolerance," in Proc. 3rd USENIX Symp. Operating Systems Design and Implementation (OSDI), New Orleans, LA, USA, Feb. 1999, pp. 173-186.

[16] D. Ongaro and J. Ousterhout, "In Search of an Understandable Consensus Algorithm," in Proc. USENIX Annual Technical Conf., Philadelphia, PA, USA, June 2014, pp. 305-319.

[17] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT Consensus with Linearity and Responsiveness," in Proc. ACM Symp. Principles of Distributed Computing (PODC), Toronto, Canada, July 2019, pp. 347-356.

[18] S. Thomas and E. Schwartz, "A Protocol for Interledger Payments," Interledger White Paper, 2015.

[19] M. Herlihy, "Atomic Cross-Chain Swaps," in Proc. ACM Symp. Principles of Distributed Computing (PODC), Egham, UK, July 2018, pp. 245-254.

[20] E. Ben-Sasson, A. Chiesa, C. Garman, et al., "Zerocash: Decentralized Anonymous Payments from Bitcoin," in Proc. IEEE Symp. Security and Privacy (SP), Berkeley, CA, USA, May 2014, pp. 459-474.

[21] S. Noether and A. Mackenzie, "Ring Confidential Transactions," Ledger, vol. 1, pp. 1-18, Dec. 2016.

[22] G. Maxwell, "Confidential Transactions," Bitcoin Core Development, Technical Note, 2016.

[23] P. Tasca, T. Thanabalasingham, and C. J. Tessone, "Ontology of Blockchain Technologies: Principles of Identification and Classification," IEEE Computer, vol. 52, no. 11, pp. 58-67, Nov. 2019.

[24] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou, "A Survey of Distributed Consensus Protocols for Blockchain Networks," IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 1432-1465, Second Quarter 2020.

[25] A. Mehra and T. Singh, "Scalability Challenges in Blockchain-Based Payment Systems," J. Financial Technology, vol. 6, no. 1, pp. 12–29, 2020.