Hybrid Blockchain-Based Tokenization for Secure and Interoperable Electronic Payment Systems

Omniah Abdullah Al Ibrahim, Suhair Alshehri
Information Technology Department-Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah 21589, Saudi Arabia

***Abstract*—The rapid evolution of digital payment technologies has highlighted critical challenges in ensuring data security, transparency, and compliance across centralized tokenization systems. Traditional models such as those used by EMVCo and major card networks rely on Token Service Providers (TSPs) and token vaults, which, while effective at masking Primary Account Numbers (PANs), still suffer from single points of failure, limited auditability, and potential privacy risks. Fully decentralized blockchain-based payment models, on the other hand, often face latency, scalability, and regulatory compliance challenges that hinder real-world adoption. This paper proposes a blockchain-based distributed tokenization model that decentralizes the tokenization and validation phases while maintaining centralized authorization and settlement through existing ISO 8583 payment rails. Unlike traditional centralized tokenization architectures that rely on a central TSP for token management, and fully decentralized approaches that migrate payment processing onto the blockchain, the proposed model selectively decentralizes token lifecycle functions while preserving compatibility with existing payment infrastructure. The model introduces a two-layer architecture: an on-chain layer for token creation, validation, and consumption using smart contracts deployed on the Ethereum network, and an off-chain layer for detokenization, authorization, and regulatory compliance through the issuer-side Token resolution service (TRS) and Token Vault. Experimental results show a 65–70% reduction in gas consumption and lower transaction latency compared to a fully on-chain reference model. These improvements stem from the model's hybrid structure, which minimizes state changes and optimizes network resource utilization. The findings indicate that the proposed model enhances security, interoperability, and operational efficiency while supporting regulatory compliance and blockchain-based token lifecycle management within real-time payment workflows, while keeping latency-critical authorization and settlement off-chain.**

***Keywords*—Blockchain-based payment systems; decentralized tokenization; electronic payment security; smart contracts; hybrid payment model**

I. Introduction

Digital payments have transformed financial transactions, and tokenized transactions accounted for over 40% of non-cash transactions on major card networks in 2024 [1]. This shift is driven by the need to reduce the risks associated with the storage and transfer of Primary Account Numbers (PANs), which can help merchants reduce the scope of PCI DSS compliance. However, the dominant centralized tokenization model, operated by a limited number of large Token Service Providers (TSPs), is built on secure vaults that maintain a two-way mapping between tokens and underlying PANs. This concentration of sensitive mappings creates high-value, high-risk targets: a single successful breach can lead to millions of payment credentials being compromised immediately. For instance, a 2024 forensic report documented a breach involving 87 million tokenized accounts [1]. At the same time, the distributed ledger technology has evolved into a promising financial application infrastructure. The transition to Proof-of-Stake in 2022 reduced Ethereum's energy consumption by more than 99.95% while preserving Ethereum's consensus-based security guarantees, making it a more practical platform for programmable financial logic [2].

Later developments in scaling solutions, including optimistic and zero-knowledge rollups, have reduced confirmation latency in some settings and made blockchain-based systems more suitable for latency-sensitive financial workflows [2]. Complex, minimized trust workflows such as conditional execution, event-driven auditing and cryptographically verifiable consent are now possible using smart contracts, and can immediately apply to the lifecycle management of payment tokens.

In spite of these developments, comprehensive on-chain payment systems cannot yet be used in the real world due to the operational constraints. Contactless point-of-sale (POS) payments require sub-500 ms authorization latency, real time fraud detection, and strict compliance with regulatory requirements (e.g. KYC, AML, and chargeback liability) that cannot be assigned to permissionless networks by issuers [3]. In addition, the international payment system is still based on the ISO 8583 messaging standards, which handle more than 180 billion card transactions per year with well-established acquiring and issuing rails [4]. This paper presents a blockchain-based distributed tokenization model that strategically decouples auditability from authorization. The system mitigates long-standing centralized vault vulnerabilities by storing only privacy-preserving metadata, including salted hashes of PAN and device identifiers, EIP-712 structured signatures, and lifecycle events (TokenIssued, TokenValidated, TokenRevoked), on the Ethereum Sepolia test network for prototype evaluation, while low-latency authorization is maintained through existing financial rails.

All sensitive data and functions that are compliance-critical are kept within Hardware Security Modules (HSMs) that are highly controlled by issuers, so they are designed to align with PCI DSS v4.0, the EMVCo Tokenization Framework, and relevant payment regulatory requirements including the European Union's Markets in Crypto-Assets Regulation (MiCA), which governs crypto-asset markets, and the revised Payment Services Directive (PSD3), which regulates electronic payment services and fraud prevention [3]. This multi-tiered structure is consistent with recent regulatory guidance that views hybrid

frameworks as a practical approach for introducing distributed systems into controlled payment environments [3].

The main contribution of this paper is the design and development of a hybrid architecture that decentralizes only the tokenization layer (issuance and audit records stored on Ethereum), while keeping authorization, fraud logic, and settlement within issuer/card network control. This separation provides a balance of transparency and efficiency. Compared with prior decentralized proposals, our system introduces:

- Explicit NFC/ISO 8583 mappings, enabling compatibility with legacy payment rails.
- A minimal on-chain data model, ensuring no PAN or personally identifiable information (PII) is exposed.
- And Device-bound consent mechanisms, leveraging EIP 712 for cryptographic request binding and WebAuthn for hardware-based key protection.

This design provides security and auditability mechanisms while maintaining practical compatibility with real-world financial ecosystems. The novelty of the proposed model lies in this selective decentralization boundary, where blockchain is used specifically for auditable token lifecycle management, while latency-sensitive authorization and settlement remain within existing ISO 8583 payment infrastructure.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the proposed hybrid tokenization model. Section IV presents the implementation and evaluation results. Finally, Section V concludes the paper and discusses future work.

II. Related Work

This section reviews the most relevant studies addressing the electronic payment tokenization systems. The reviewed works are classified into two main categories: (1) centralized tokenization approaches, which rely on traditional token service providers (TSPs) and secure vaults, and (2) decentralized or distributed ledger technology (DLT)-based approaches, which employ blockchain to enhance transparency and auditability.

A. Centralized Tokenization Approaches

The increasing use of mobile payments has increased the need to protect credit card information against physical and cyber threats.

In [5], the authors proposed an offline virtual credit card generation system based on ARM TrustZone technology to enhance mobile payment security. The system operates within the device's Trusted Execution Environment (TEE), ensuring that sensitive credentials are isolated from the Rich Execution Environment (REE), which is more vulnerable to malware and external attacks. By enabling local virtual card generation without requiring real-time internet connectivity, the approach reduces exposure to remote interception and server-side attacks. Cryptographic synchronization between the device and issuing bank further enhances integrity and consistency. However, the solution is heavily dependent on hardware-level TrustZone support, limiting scalability and interoperability across diverse device ecosystems. Additionally, the lack of continuous real-time synchronization with financial institutions may weaken fraud detection and compliance monitoring.

In [6], the authors introduced an Upcycling Tokenization Method designed to improve token lifecycle management efficiency in centralized payment systems. Unlike conventional tokenization approaches that continuously generate new tokens, the proposed model recycles expired tokens using a dynamic and memory-optimized mapping structure. This optimization reduces storage overhead and improves system performance under moderate transaction loads. The design enhances operational efficiency while maintaining compatibility with existing Token Service Provider (TSP) infrastructures. Nevertheless, the architecture remains dependent on a centralized TSP for token issuance and validation. Consequently, it preserves the inherent risks of single points of failure, insider threats, and vault compromise, limiting systemic resilience.

In [7], the authors proposed a secure credit card payment model tailored for e-platforms, integrating tokenization with symmetric encryption and multi-factor authentication (2FA). The system replaces sensitive cardholder data with surrogate tokens and encrypts transactional data using XOR operations combined with the Advanced Encryption Standard (AES). The layered security approach enhances confidentiality and strengthens authentication mechanisms in online environments. However, the model still relies on centralized token management and trusted intermediaries for encryption key storage and token verification. This centralization introduces potential vulnerabilities, including insider threats and infrastructure-level attacks. Furthermore, the combined use of tokenization and encryption may introduce processing overhead under high transaction volumes.

In [8], the authors proposed a token-based secure payment system utilizing SHA-256 cryptographic hashing to protect card-based transactions in mobile and cloud environments. The framework replaces Primary Account Numbers (PANs) with tokens and applies irreversible hashing to enhance confidentiality. The system integrates with EMV standards and supports Host Card Emulation (HCE) and Near Field Communication (NFC) technologies, ensuring compatibility with existing POS infrastructures. This design strengthens resistance against replay attacks and unauthorized data modification. Despite these improvements, the architecture remains centralized and dependent on a Token Service Provider for token lifecycle management. As a result, independent auditability is limited, and systemic risks associated with centralized vault breaches persist.

In [9], the authors addressed vulnerabilities in EMV contactless payment cards by proposing a tokenization-based replacement of Primary Account Numbers during transaction processing. The system embeds tokenization within the card personalization phase, allowing tokens to be transmitted instead of actual PANs during contactless interactions. The approach improves privacy and reduces exposure to NFC skimming and cloning attacks while maintaining compatibility with existing EMV infrastructures. However, the system does not eliminate advanced relay attacks and continues to rely on centralized infrastructure for token management. Consequently, transparency remains limited, and systemic risks related to centralized trust entities are not fully mitigated.

TABLE I. FEATURE-BASED EVALUATION OF TOKENIZATION APPROACHES

Researches/ Features	Enhanced Privacy & Security	Merchant Flexibility & Compatibility	Issuer Control	Incremental Adoption	Regulatory Simplicity	High Performance	Reduced Attack Surface
[5]	✓	✗	✗	✗	✓	✓	✗
[6]	✗	✓	✓	✓	✓	✓	✗
[7]	✗	✓	✓	✓	✓	✗	✗
[8]	✗	✓	✓	✓	✓	✓	✗
[9]	✗	✓	✓	✓	✓	✓	✗
[10]	✓	✗	✗	✗	✗	✗	✓
[11]	✓	✗	✗	✗	✗	✗	✗
[12]	✓	✗	✗	✗	✗	✗	✓
Our Proposed System	✓	✓	✓	✓	✓	✓	✓

Collectively, centralized tokenization approaches enhance confidentiality through encryption, hardware isolation, and token substitution. However, they consistently retain structural dependencies on centralized Token Service Providers, creating persistent risks such as single points of failure, limited transparency, and constrained independent auditability. These limitations highlight the need for tokenization models that improve auditability and resilience without sacrificing interoperability with existing payment infrastructures.

B. Decentralized Tokenization Approaches

Research in decentralized tokenization explores the limitations of centralized tokenization. These models use distributed ledgers and smart contracts to manage tokens without relying entirely on centralized intermediaries. Their goal is to enhance security, privacy, transparency, and auditability while addressing single points of failure and insider threats.

In [10], the authors proposed a blockchain-based payment system utilizing Non-Fungible Tokens (NFTs) as identifiable and verifiable payment credentials within an academic financial context. The model operates on a permissionless Ethereum network and stores metadata on the InterPlanetary File System (IPFS), enhancing decentralization and immutability. By replacing traditional token vaults with blockchain-based representations, the system improves transparency, traceability, and resistance to insider manipulation. However, reliance on public blockchain infrastructure introduces scalability limitations, transaction latency, and high gas costs. These constraints reduce suitability for real-time retail POS environments.

In [11], the authors introduced a distributed token-based payment framework built on a private consortium blockchain. The system assigns participating financial institutions—such as issuer banks and acquirer banks—to dedicated sub-blockchains governed by a central authority. This architecture enhances auditability and privacy for card-not-present transactions while restricting access to sensitive payment data through controlled token exchange mechanisms. Nevertheless, the hybrid governance model preserves centralized coordination components, particularly within the supervising authority. Additionally, token-to-PAN mappings are still maintained in controlled vault environments, meaning certain centralization risks remain unresolved.

In [12], the authors proposed a fully decentralized blockchain-based tokenization framework implemented on the Ethereum network. The system replaces traditional Token Service Providers with smart contracts that autonomously generate, validate, and manage token lifecycles. Blockchain immutability ensures transparent, tamper-resistant transaction logging, and the elimination of intermediaries reduces insider risk. The authors reported deployment gas, execution gas, and latency metrics to evaluate system performance. Despite its strong decentralization properties, the framework faces significant scalability challenges due to network congestion, gas cost volatility, and blockchain confirmation delays. These operational constraints limit its applicability in high-volume, real-time retail payment settings. Decentralized tokenization models improve transparency, resilience, and auditability by eliminating centralized intermediaries. However, their dependence on blockchain consensus mechanisms introduces latency, cost variability, and regulatory integration challenges. These limitations reveal a trade-off between architectural decentralization and operational practicality.

C. Comparative Analysis of Tokenization Approaches

The reviewed studies exhibit significant diversity in scope and design, particularly concerning the degree of decentralization, privacy mechanisms, regulatory alignment, and performance optimization. Table I compares these works according to their core characteristics, providing a comprehensive view of how existing solutions address privacy, issuer control, merchant compatibility, and overall system efficiency.

The comparison is based on the following key features identified across the literature:

- Enhanced Privacy and Security: Physical Point-of-Sale (POS) terminals will never expose, transmit or store PAN and sensitive data.
- Merchant Flexibility and Compatibility: The system is easily integrated with the current POS infrastructure and card networks.
- Issuer Control: Fraud detection, funds authorization, and compliance logic are fully controlled by issuers.
- Incremental Adoption: Stakeholders do not need to make disruptive changes to the whole payment stack to be able to adopt the decentralized token layer.
- Regulatory Simplicity: The complexities of regulation are reduced because decentralization will be restricted to tokenization rather than full decentralization.
- High Performance: Settlement and authorization activities preserve the low-latency of traditional systems.
- Reduced Attack Surface: The reliance on centralized Token Service Providers (TSPs) creates a single point

of failure, which some approaches address by partially decentralizing token management.

These comparative criteria provide a structured basis for evaluating the strengths and limitations of existing tokenization frameworks within electronic payment systems. The proposed system aims to address these limitations by combining decentralized token lifecycle management with compatibility with existing payment infrastructure.

III. The Proposed Model

This section presents the proposed Blockchain-Based Distributed Tokenization model that integrates blockchain-based tokenization with existing payment-network infrastructures. The model aims to achieve secure, privacy-preserving, and interoperable electronic payments without disrupting legacy ISO 8583 messaging standards or EMV workflows. The section outlines the system's architectural components, data structures, and operational processes across on-chain and off-chain domains. It further describes the mapping of tokenized transactions into ISO 8583 message fields to maintain compatibility with current acquiring and issuing systems.

The proposed model is designed to address the security and operational limitations of conventional centralized tokenization frameworks, which often suffer from single points of failure and restricted interoperability. By introducing a hybrid on-chain/off-chain tokenization structure, the model decentralizes the token generation and verification phases while retaining compatibility with established ISO 8583 and EMV infrastructures. This design enhances confidentiality, reduces systemic risks, and supports incremental adoption by financial institutions without disrupting existing payment workflows.

A. System Components and Roles

The proposed model involves five principal actors operating across four logical layers. These layers: Wallet Layer, On-Chain Tokenization Layer, Payment Rails Layer, and Token Resolution Service Layer work together to provide security, privacy, auditability, and compatibility with NFC-enabled point-of-sale (POS) terminals and existing payment networks. Each actor's role, functionality, and standards alignment are summarized below. Fig. 1 shows the system actors and layers in the proposed model. The diagram illustrates the primary entities involved in the proposed system: the cardholder wallet, merchant POS, acquirer gateway, and Token Resolution Service (TRS). Each actor operates within a specific layer user, merchant, network, or issuer interacting through tokenized ISO 8583 messages. The figure emphasizes the layered design that separates token management from payment authorization to enhance interoperability and security [13], [14], [15], [16].

1) System actors: The system involves five principal actors: Cardholder Wallet, Merchant POS, Acquirer Gateway, TRS, and Ethereum Tokenization Contracts.

- Cardholder Wallet (Mobile/Web):
  A user-facing application enables cardholders to initiate payments, create token requests, and sign them cryptographically. The cardholder wallet maintains user control of payments while preventing exposure of Primary Account Numbers (PANs).

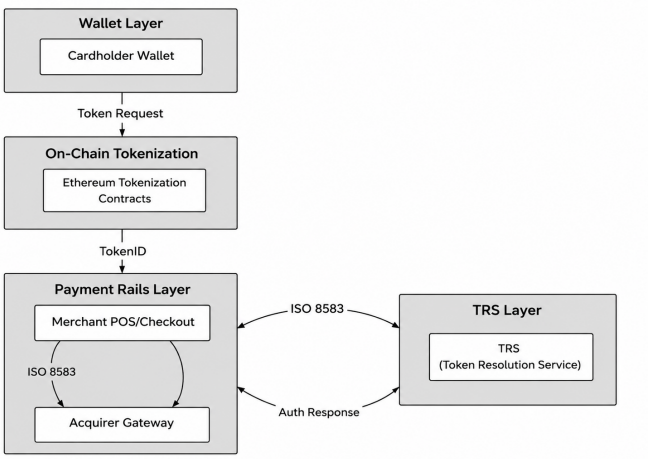

Fig. 1. System actors and layers.

  - Functions:
    - Manages Keys: Stores private keys in secure hardware such as a Trusted Execution Environment (TEE) or Secure Enclave to prevent key leakage.
    - Generates Tokens: Requests one-time-use tokens from the issuer for each payment session.
    - Encrypts Data: Protects sensitive card data and transaction details using symmetric encryption before transmission.
    - Signs Requests: Uses EIP-712 compliant message signing to guarantee integrity and authenticity of payment requests.
    - Stores Metadata: Keeps non-sensitive transaction references locally for audit and offline validation.
    - Interfaces with POS: Communicates through NFC or QR channels with merchant terminals for token exchange.
  - Standards Compliance: WebAuthn and EIP-712 [17], [18]; EMV [13].

- Merchant POS:
  Existing NFC-enabled POS terminals or online checkouts accept tokenized payments and forward them to the acquiring bank using ISO 8583 messages. Merchant POS utilizes merchants' current infrastructure while reducing PCI DSS scope.
  - Functions:
    - Receives token IDs from the wallet (via NFC or web payload).
    - Packages transaction data in ISO 8583 fields (e.g., DE48/DE55) while leaving DE2 (PAN) empty.
    - Forwards token-based messages to the acquirer gateway.
  - Standards Compliance: ISO 8583 [15], EMV tokenization [13].

- Acquirer Gateway:

Middleware that relays ISO 8583 messages between merchants and issuers without accessing PANs. Acquirer gateway maintains the conventional payment-network workflow while supporting tokenized transactions.
    - Functions: Receives token-based ISO 8583 messages, forwards them to the issuer, and returns authorization responses.
    - Standards Compliance: ISO 8583 [15].

- TRS:
A PCI-compliant issuer-operated service that stores PANs securely and resolves tokens for authorization. The TRS ensures sensitive data remain within a controlled environment and enables fraud detection. The proposed architecture does not aim to eliminate centralized issuer functions. The TRS remains a trusted issuer-operated component responsible for PAN storage, token resolution, fraud checking, and payment authorization. Accordingly, decentralization is applied to on-chain token lifecycle management and auditability, while sensitive, regulatory, and latency-critical payment functions remain under issuer control.
    - Functions:
        - Stores PANs in a Hardware Security Module (HSM) and generates salted hashes for on-chain references.
        - Maps token IDs to transaction data for authorization.
        - Performs fraud checks and updates token status to prevent reuse.
    - Standards Compliance: PCI DSS [14], EMV tokenization [13].

- Ethereum Tokenization Contracts (ETC):
Smart contracts on the Ethereum blockchain issue, validate, and audit dynamic tokens without storing PANs or personally identifiable information. These contracts provide decentralized, immutable token management and public auditability.
    - Functions:
        - Issues tokens upon verification of EIP-712-signed requests.
        - Validates token scope, expiry, and status.
        - Emits events for transparent logging e.g., TokenIssued, TokenValidated.
    - Standards Compliance: Ethereum smart-contract standards [19], EIP-712 [19].

2) Layered architecture: The system involves four logical layers. These layers are Wallet Layer, On-Chain Tokenization Layer, Payment Rails Layer, and TRS Layer.

- L1 – Wallet Layer: Handles user interaction, token creation, and signing via EIP-712 and WebAuthn for secure, device-bound authentication.
- L2 – On-Chain Tokenization Layer: Executes Ethereum smart contracts to issue and validate tokens while storing only minimal metadata.
- L3 – Payment Rails Layer: Transmits tokens using ISO 8583 messaging, preserving compatibility with existing financial networks.
- L4 – TRS Layer: Performs off-chain token-to-PAN mapping and authorization within a PCI-DSS-compliant environment.

B. Data Model

The proposed system employs a dual data model to balance privacy, auditability, and regulatory compliance. Minimal transaction metadata is stored on-chain, while sensitive details remain off-chain in a PCI-compliant environment.

1) On-chain (ethereum tokenization contracts): The on-chain schema is intentionally minimal to avoid exposing personally identifiable information (PII). The token is constructed using the following parameters.

```
1. struct Token {
2. address issuer;   // who minted the token
3. bytes32 customerId;  // hashed user identifier
4. bytes32 merchantId;    // hashed merchant identifier
5. uint96 amountMinor;  // example: 100.25 SAR → 10025 (integer minor units)
6. uint64 expiry;    // unix time in seconds
7. Status status;    // token state: Active, Spent, Revoked
```

Fig. 2. On-chain token data model.

Fig. 2 presents the Solidity-based structure used to define tokens within the Ethereum smart contract. Each token record includes parameters such as tokenId, merchantRef, amountMinor, expiry, and status, linked through a hash reference to ensure integrity and prevent duplication. The minimalist schema reduces on-chain data exposure while preserving auditability and compliance with PCI DSS and EMVCo tokenization guidelines.

Token expiry is enforced by the smart contract's validate() function, which rejects any transaction where block.timestamp exceeds the token's expiry value. The issuer's TRS performs a synchronized expiry check to prevent stale token authorization.

Events

- TokenIssued: Records token creation.
- TokenValidated: Records successful validation by an issuer.
- TokenRevoked: Records token revocation with reason codes.

Design Rationale

- cardRef is computed as keccak256(salt ‖ PAN ‖ deviceId) inside the issuer's Hardware Security Module (HSM). Prevents PAN disclosure while maintaining a stable reference [14].
- requestHash follows EIP-712 structured data signing, binding user consent to transaction parameters for non-repudiation [19].
- Storing only essential metadata (no PAN/PII) lowers gas costs and mitigates privacy risk compared with NFT-style token systems [10].

2) Off-chain (TRS): Sensitive data and cryptographic credentials are retained within the issuer's PCI-scoped infrastructure:

- Token Vault: Maintains the authoritative mapping of tokens to PANs and transaction state. All PANs and salts remain in HSM storage [14]. It stores tokenId, cardRef, PAN (HSM), merchantId, amount, currency, expiry, status, attempts.
- KeyStore: Stores wallet public keys, WebAuthn attestations, and issuer/acquirer TLS certificates. Enables mutual authentication and secure communication. Preserves cryptographic credentials and supports WebAuthn validation [18].

This partitioned design ensures that public blockchain records are auditable while all regulated payment data remains protected within a PCI-DSS–compliant boundary.

C. Functions

The system supports four primary processes: customer registration, merchant registration, token creation/verification, and off-chain payment. These functions ensure secure, privacy-preserving transactions across card-present (NFC) and card-not-present (web) scenarios.

1) Customer registration: This process enrolls the customer's device and card while ensuring that Primary Account Numbers (PANs) remain within the issuer's PCI-controlled environment. Fig. 3 shows the customer registration process. The figure illustrates how a customer enrolls a payment device and card into the issuer's environment while preserving Primary Account Number (PAN) confidentiality. The customer wallet submits an encrypted registration request containing device attestation and card alias. The issuer performs KYC/AML verification, validates the WebAuthn attestation, and generates a unique customerId. A secure cardRef = keccak256(salt —— PAN —— deviceId) is computed inside the Hardware Security Module (HSM), ensuring that the PAN never leaves the PCI-controlled boundary. Optionally, a hash of the customer reference may be recorded on-chain for auditability. The issuer stores all sensitive data in the TokenVault and returns non-sensitive identifiers to the wallet for future tokenization [13], [14], [18].

Process

- Registration Request – The customer wallet submits legalName?, email?, deviceAttestation, cardAlias through a TLS-secured API.
- Verification – The issuer performs KYC/AML checks and validates WebAuthn attestation to confirm device authenticity.
- Customer-ID Creation – A unique customerId is generated, and a salted hash cardRef = keccak256(salt —— PAN —— deviceId) is computed inside a Hardware Security Module (HSM).
- Blockchain Record (Optional) – A privacy-preserving hash of the customer reference can be stored on-chain for auditability.
- Secure Storage – The issuer records customerId, cardRef, PAN (HSM-encrypted), status=Active in the TokenVault.
- Response – The wallet receives customerId and cardRef for subsequent token generation.

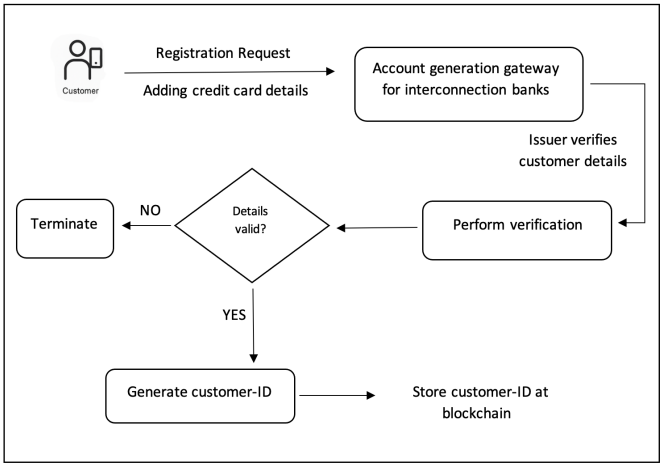

Fig. 3. Customer registration process.

2) Merchant registration: To bind a merchant verified off-chain identity to an on-chain reference used for token scoping. Fig. 4 shows the merchant registration process. The figure depicts the onboarding of merchants into the hybrid tokenization framework. A merchant submits business and payment-service credentials through a secure portal or API. The registrar or acquirer performs compliance and risk checks before issuing a unique merchantId. The smart contract then maps this merchantId to a hashed merchantRef within the on-chain TokenRegistry, establishing a verifiable identity for token scoping. Simultaneously, the acquirer's database maintains a merchantId $\leftrightarrow$ merchantRef mapping to ensure ISO 8583 compatibility during payment processing. This dual on-chain/off-chain registration preserves authenticity while maintaining backward compatibility with existing acquirer systems [13], [15].

Process

- Data Submission – The merchant provides business and payment-service details via a secure portal or API.
- Verification – The registrar performs risk checks and issues a unique merchantId.
- On-Chain Mapping – A smart contract call links merchantId to a hashed merchantRef on the TokenRegistry.
- Database Mapping – The acquirer maintains a merchantId $\leftrightarrow$ merchantRef mapping for ISO 8583 compatibility.

3) Token generation and validation (on-chain): To create dynamic, single-use payment tokens and validate them before authorization. Fig. 5 shows token generation and validation flow in the proposed model. This flowchart demonstrates how a transaction request travels from the merchant POS to the issuer

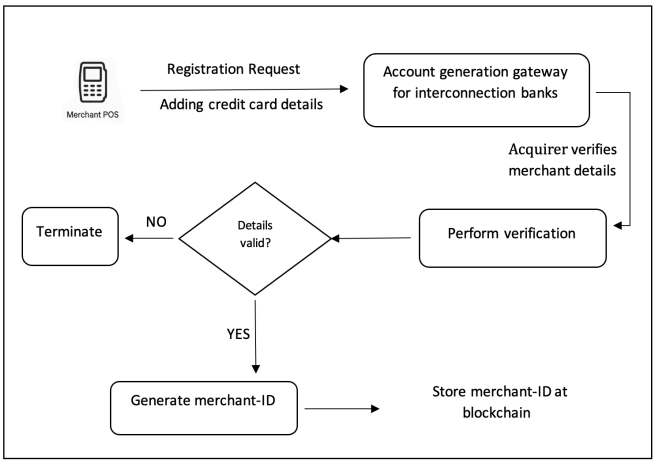

Fig. 4. Merchant registration process.

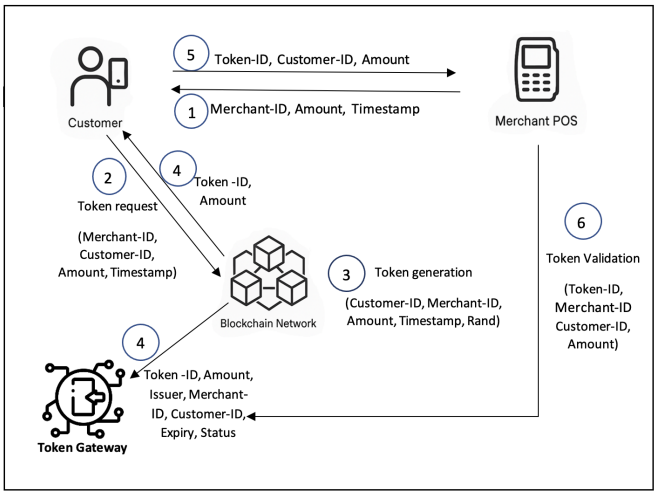

Fig. 5. Token generation and validation flow in the proposed model.

through the blockchain network. The customer wallet creates a token request containing identifiers and transaction details; the blockchain generates a unique token and records it immutably on-chain. When the merchant POS submits the payment, the acquirer gateway transmits the ISO 8583 authorization message to the issuer. The issuer's TRS then performs on-chain token validation by verifying the token's authenticity, expiry, and status before detokenization and authorization. This design enables real-time validation and ensures that replayed or duplicated tokens cannot be reused, preserving both integrity and interoperability with existing payment rails [13], [14], [15], [20].

Token Generation Flow

- Token Request – The wallet forms a TokenRequest containing cardRef, merchantRef, amountMinor, currency, timestamp, and a random nonce.
- EIP-712 Signing – The request is signed with the wallet's private key, ensuring non-repudiation.
- Issuance – The signed request is submitted to the TokenFactory smart contract.
- Contract Actions – The contract verifies the signature, checks freshness, generates a unique tokenId, stores TokenMeta, and emits a TokenIssued event.
- Response – The caller receives tokenId, amountMinor, and issuer details.

Token Validation Flow

- Validation Call – The issuer invokes validate (tokenId, merchantRef) on-chain.
- Checks Performed – The contract verifies token existence, expiration, merchant scope, cardRef match, and status.
- Outcome – Returns a Boolean result and emits a TokenValidated event.

4) Payment using token (off-chain): To complete payments over existing card-network rails while preserving PAN privacy. Fig. 6 shows the payment using token process. The diagram demonstrates how both card-present (NFC) and card-not-present (web) payments are executed using blockchain-issued tokens instead of PANs. In NFC transactions, the wallet transmits the tokenId to the POS terminal, which embeds it along with amount and merchant data into an ISO 8583 message where the PAN field is intentionally left empty. The acquirer gateway routes the message unchanged to the issuer, whose TRS detokenizes and authorizes the transaction inside an HSM-protected TokenVault. For web checkouts, the same process occurs through a 3-D Secure or gateway payload. After successful authorization, funds are settled through existing rails, and the token's state is updated via markSpent or revoke to ensure single-use compliance and auditability [14], [15], [12].

Process

Card-Present (NFC)

- The wallet transmits tokenId to the POS terminal via NFC.
- The POS embeds tokenId, amountMinor, currency, and merchantId in an ISO 8583 message with the PAN field left empty.
- The acquirer routes the message to the issuer for authorization.

Card-Not-Present (Web)

- During checkout, the wallet SDK signs and issues a token which is inserted into a 3-D Secure or gateway payload.
- The acquirer forwards the payload to the issuer.

Issuer/Processor Actions

- Detokenization: Securely maps tokenId to PAN, amount, currency, merchantId inside the HSM-protected TokenVault.
- Authorization: Performs fraud/risk checks and returns APPROVED/DECLINED via ISO 8583.

- Settlement & Confirmation: Funds are transferred through standard clearing; markSpent or revoke may be invoked to finalize token status.

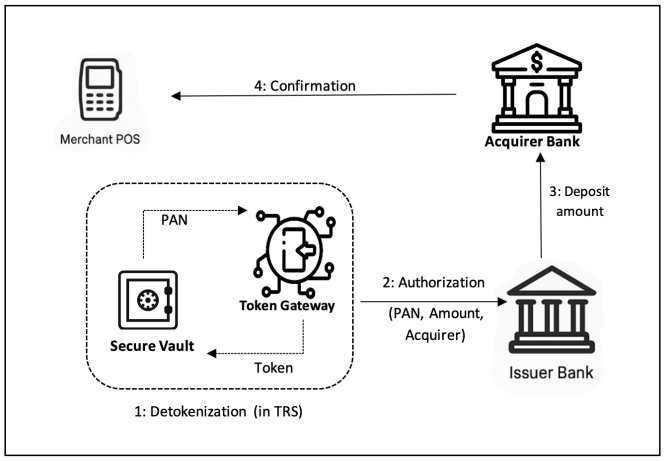

Fig. 6. Payment using token process.

5) Reversal or refund: To manage transaction reversals or customer-requested refunds with auditability.

Process

- The merchant submits the original tokenId and amount to the acquirer.
- The issuer verifies the TokenVault record and processes the reversal.
- On-chain, revokeToken(tokenId, reason) updates the token's status and emits a TokenRevoked event.

This structure presents the complete life-cycle from customer registration to token issuance, validation, payment processing, and post-transaction management in a format consistent with academic reporting standards. Upon successful reversal, the TRS emits a RefundProcessed event to the merchant, and the corresponding on-chain record is optionally updated with a revokeToken call to preserve an immutable audit trail.

The wallet, POS/checkout adapter, and issuer-side TRS interact through signed token requests, ISO 8583-compatible payment messages, and synchronized token-state updates between the on-chain contract and the off-chain TokenVault.

D. ISO 8583 Mapping

After defining the functional processes and interactions among the wallet, merchant, acquirer, and issuer layers, it is essential to demonstrate how the proposed tokenized transaction can traverse conventional payment networks without structural disruption. To achieve this, the model incorporates a standardized message translation mechanism that aligns token-related data with the ISO 8583 financial message format used across global card-payment systems. This mapping ensures that newly introduced fields such as token identifiers and on-chain references fit within existing message specifications while preserving interoperability, regulatory compliance, and data confidentiality.

ISO 8583 is the standardized message format used globally for financial transaction authorization and settlement. So, ISO 8583 Mapping defines how our system's internal fields (token ID, cryptographic proofs, blockchain references) are translated into ISO 8583 Data Elements (DEs), ensuring interoperability with existing rails while maintaining security and privacy.

- DE2 (PAN): Intentionally left blank to protect cardholder data [14]. Avoids transmitting or storing the actual PAN; instead, a token or reference is used off-chain, protecting sensitive cardholder information.
- DE3, DE4, DE49: Standard processing code, amount, and currency for interoperability [15]. Ensure compatibility with standard payment networks; these are mandatory for transaction routing.
- DE41/DE42: Terminal and merchant identifiers. Identify where and by whom the transaction occurred essential for acquirers and issuers to trace transactions.
- DE48/DE55: Carry tokenId and optional on-chain receipt hash for auditing [15], [21]. Leveraged for modern tokenization and blockchain audit data, such as token identifiers, expiry time, or the hash of the on-chain transaction receipt, enabling transparency and forensic validation.

This mapping ensures seamless routing through existing card-payment networks while preserving privacy. The ISO 8583 mapping layer serves as the critical bridge between the decentralized tokenization environment and the traditional financial infrastructure. By embedding token identifiers, cryptographic proofs, and optional blockchain audit references into existing ISO 8583 fields, the proposed model maintains compatibility with established acquiring and issuing networks while eliminating exposure of primary account numbers (PANs). This approach enables seamless message routing and settlement through legacy rails, enhances transparency through verifiable records, and upholds the confidentiality and integrity required by PCI-DSS and EMV standards. The next section further elaborates on how these design choices collectively strengthen the system's overall security, privacy, and compliance posture.

E. Security, Privacy, and Compliance

Building on the interoperability achieved through ISO 8583 mapping, this section outlines the mechanisms that preserve the model's security, privacy, and compliance across all operational layers. Key safeguards include:

- PAN Confidentiality: Only cardRef and tokenId circulate; PAN remains inside HSM-protected issuer systems [14].
- Replay & Scope Control: nonce and merchantRef enforce one-time, time-bound, merchant-specific tokens; EIP-712 signing binds consent [19].
- Transport Security: TLS plus optional AES-GCM encryption protects data in transit [22].
- Auditability: Blockchain events (TokenIssued, TokenValidated, TokenRevoked) and vault logs provide a verifiable history [12].

- Device Security: WebAuthn keys remain non-exportable and hardware-attested [18].
- Standards Compliance: Conforms to EMV [13], PCI DSS [14], and ISO 8583 [15].

IV. Implementation and Results

This section presents the implementation of the proposed model in Section IV-A, and the experimental results and comparative analysis in Section IV-B.

A. Implementation

We successfully implemented the proposed Distributed Tokenization System on a system running Windows 10 Home Single Language 22H2 with an Intel(R) Core (TM) i5-4200M CPU @ 2.50 GHz and 6.0 GB of RAM. The Remix IDE was used to develop and deploy smart contracts on the public Ethereum Sepolia test network. Remix operates entirely within a web browser, eliminating the need for local configuration. The smart contracts were written in Solidity and deployed on the Ethereum Sepolia test network using Remix IDE, while MetaMask was used for account creation and transaction initiation. The TRS component was simulated using the Replit cloud-based IDE to emulate off-chain operations.

The implementation is divided into on-chain and off-chain components. The on-chain contracts perform registration, tokenization, validation, and consumption, while the off-chain layer handles detokenization and authorization using ISO 8583-compliant gateways.

The on-chain layer consists of two Solidity smart contracts: a Registry contract and a Tokenization contract. The Registry contract registers customers and merchants using only hashed identifiers, while the Tokenization contract manages token generation, validation, consumption, and revocation. The off-chain TRS simulates detokenization and ISO 8583-compatible authorization by mapping token identifiers to protected payment credentials and invoking the corresponding on-chain lifecycle functions.

The implementation aims to demonstrate the functional interoperability between blockchain-based token management and traditional payment infrastructure. Functional testing confirmed that token generation, validation, consumption, and revocation executed correctly on Sepolia, including the rejection of reused, expired, or inactive tokens.

B. Results and Comparative Analysis

This section evaluates the feasibility and performance of the proposed blockchain-based tokenization model through experimental analysis and comparative assessment with the closely related work in [12]. The evaluation considers deployment gas cost, transaction gas consumption, effective gas price, transaction fee, and latency.

Because the proposed and reference systems were evaluated under different experimental conditions, the comparison is based on functionally equivalent operations and is intended to provide an indicative assessment rather than a controlled performance benchmark.

Table II presents the deployed smart contracts and their corresponding deployment gas costs on the Ethereum Sepolia test network, providing a foundational reference for evaluating the system's computational efficiency and operational feasibility.

TABLE II. Smart Contract Deployment Addresses and Gas Cost

Smart Contract	Deployment Address	Deployment Gas Cost
SC_ Registry	0x42671Ae42c8297075Db7F7536b8C0462121f7989	50,353
SC_ Tokenization	0x75dB5d304DE419B7C4eC26f12159AA4f2FC2054e	1,630,491

The evaluation of the proposed model is based on several key performance metrics: gas consumption, effective gas price, transaction fee, and transaction latency.

- Gas Consumption: Amount of gas consumed by the blockchain operation. Represents Ethereum computational work required for execution.
- Effective Gas Price: The actual price per unit of gas paid when the transaction is mined. This value reflects the prevailing network conditions and the miner's acceptance threshold at the time of confirmation.
- Transaction Fee: The total amount of Ether spent on executing a transaction, equivalent to the effective gas price multiplied by the gas used. This metric provides a monetary measure of computational expenditure.
- Transaction Latency: The elapsed time between the transaction submission and its inclusion in a mined block, determined by comparing the client-side submission timestamp with the blockchain block timestamp. Transaction latency varies depending on network congestion and is a critical factor in assessing the system's suitability for real-time payment processing.

Table III presents the measured gas consumption, effective gas price, transaction fee, and latency for the main system operations, including customer and merchant registration, token generation, and token consumption.

To further evaluate the proposed model, its performance was compared with the closely related Ethereum-based tokenization framework in [12]. The reference study was selected because it reports quantitative deployment gas, execution gas, and latency results for a blockchain-based tokenization system. Because the two architectures use different contract structures, the comparison is based on functional equivalence rather than identical contract naming. The proposed model uses two primary contracts, Registry and Tokenization, while the reference model uses separate registration and tokenization contracts with deposit and regulator-related mechanisms. Table IV presents the comparative deployment gas costs of the smart contracts in both systems.

As shown in Table IV, the proposed model requires lower deployment gas than the reference model, primarily because it uses a simplified Registry contract and avoids separate deposit-based registration and regulatory verification mechanisms.

The comparisons of transaction gas consumption and latency for registration, token generation, and token consumption are presented in Fig. 7–10.

Fig. 7 compares the gas consumption of customer and merchant registration between the proposed model and the

TABLE III. THE RESULT METRICS OF THE TRANSACTIONS

Process	Transaction hash	Etherscan link	Gas Consumption	Effective Gas Price (Gwei)	Transaction Fee (ETH)	Latency (Sec)
Customer/ Merchant Registration	0x83ab4037ed7cf6f67b02ab65990e518a73e56bb6d3c11d7fc6b7e796111fe572	https://sepolia.etherscan.io/tx/0x83ab4037ed7cf6f67b02ab65990e518a73e56bb6d3c11d7fc6b7e796111fe572/404	50,353	1.500000016	0.000075529500805648	4
Generate Token	0x1075a9f4494225f4c97c39df572955ad75824dd2adcccb9f825377db16fda111	https://sepolia.etherscan.io/tx/0x1075a9f4494225f4c97c39df572955ad75824dd2adcccb9f825377db16fda111	121,385	1.500031549	0.000182081329575365	10
Consume Token	0x8f939ae477ba0c6f052e3b7ae9b683aa10dbd585dc1cc56afb5a143de01f66b1	https://sepolia.etherscan.io/tx/0x8f939ae477ba0c6f052e3b7ae9b683aa10dbd585dc1cc56afb5a143de01f66b1	33,799	1.50001979	0.00005069916888221	4.5

TABLE IV. THE COMPARISON OF SMART CONTRACT DEPLOYMENT GAS

Contracts	Paper: Deployment Gas	Our Proposed: Deployment Gas
SC Bank Registration	2,267,900	(not used in our system)
SC Customer Registration	2,721,206	50,353
SC Tokenization	4,830,243	1,630,491

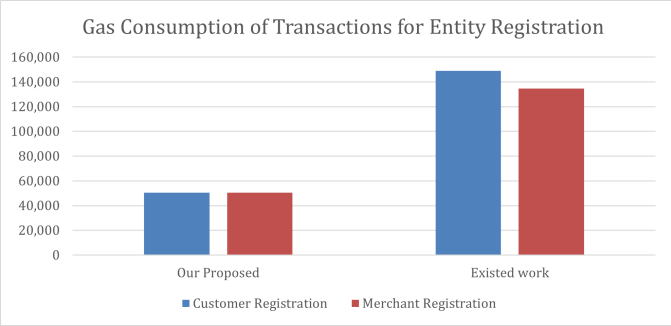

Fig. 7. Gas consumption of transactions for entity registration.

existing blockchain-based tokenization framework. The results indicate that both registration processes in the proposed model consume less gas approximately one-third of that required by the existing approach. This improvement stems from the simplified Registry contract in the proposed system, which stores only hashed identifiers (Customer-ID and Merchant-ID) without additional regulatory or deposit logic. By eliminating unnecessary on-chain data storage and complex permission checks, the system reduces computational overhead, resulting in lower gas utilization and higher cost efficiency.

Fig. 8 shows the gas consumption for token generation and token consumption processes. The proposed model demonstrates a marked reduction in gas cost particularly for token consumption, where the efficiency improvement exceeds 70%. This gain arises from the hybrid design, which keeps only essential operations (token generation, validation, and state update) on-chain while offloading detokenization and authorization to the off-chain TRS. The removal of regulatory deposits, timeout functions, and multiple contract interactions present in the reference system further reduces the number of state-changing transactions, resulting in lower gas expenditure and more scalable performance.

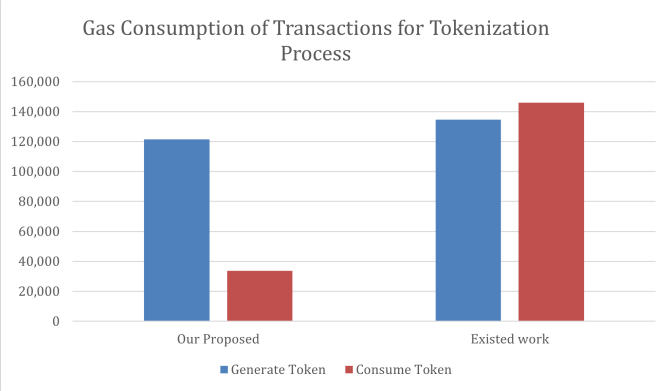

Fig. 8. Gas consumption of transactions for card tokenization process.

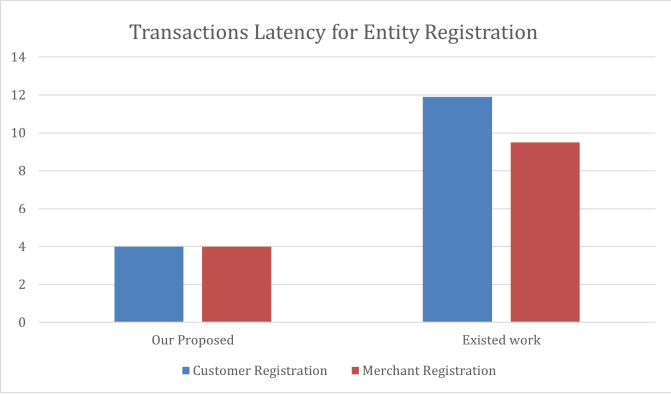

Fig. 9. Transactions latency for entity registration.

Fig. 9 compares the latency of customer and merchant registration transactions. The proposed model consistently achieves lower latency around 4 seconds compared to 10–12 seconds in the existing system. This improvement can be attributed to the smaller contract size and simplified execution path, which require fewer computational steps during block confirmation. Since only hashed identifiers are registered without additional validation layers, the transaction confirmation time decreases, enhancing real-time responsiveness for entity onboarding.

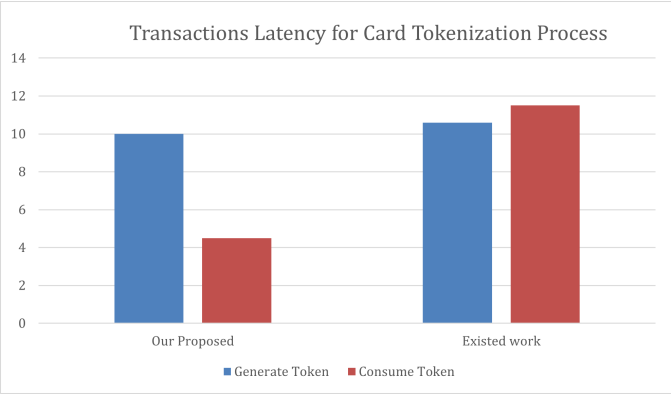

Fig. 10. Transactions latency for card tokenization process.

Fig. 10 presents the transaction latency results for token generation and consumption. The proposed system achieves reduced latency, particularly during the token consumption phase. This improvement is due to the hybrid architecture, where authorization and settlement occur off-chain through the TRS, minimizing on-chain interactions to a single state update. Additionally, the removal of complex inter-contract dependencies (such as regulator or escrow contracts) shortens execution time and simplifies transaction flow. These results indicate that the proposed model can support token lifecycle management for POS payment workflows, while real-time authorization and settlement remain off-chain within issuer and acquirer infrastructure.

The measured blockchain transaction latency of approximately 4 seconds should not be interpreted as end-to-end POS authorization latency. The reported value represents the blockchain transaction processing time under the tested conditions. Since latency-critical authorization and settlement remain off-chain through the TRS and conventional payment rails, this blockchain latency would not be suitable if blockchain confirmation were required synchronously for every POS authorization. This separation is therefore a deliberate architectural choice: blockchain is used for token lifecycle management and auditability, while the latency-critical authorization path remains within the existing payment infrastructure.

It should be noted that the reported latency measurements were obtained on the public Ethereum Sepolia test network and may therefore be affected by external network conditions, including network congestion, block-time variation. Consequently, the measured latency values should be interpreted as experimental observations under the Sepolia conditions rather than fixed execution times. Since the comparison with the reference system is based on previously reported results rather than measurements collected under identical network conditions, the latency comparison should be interpreted as indicative rather than as a controlled network-level benchmark.

The current evaluation is intended to demonstrate prototype feasibility and functional interoperability; comprehensive scalability, concurrency, and production-level real-time performance evaluation are outside the scope of this study and are considered future work.

Across the evaluated parameters, the proposed model shows lower gas consumption and observed transaction latency than the reported values of the reference system. The improvements are primarily due to 1) the separation of on-chain token management from off-chain authorization, 2) the minimalistic contract design that reduces state changes, and 3) the avoidance of redundant entities such as regulators and deposit handlers. These outcomes collectively validate the model's design objective: achieving decentralized tokenization with high efficiency and compatibility with existing POS payment infrastructure.

V. Conclusion

This study presented a blockchain-based distributed tokenization model designed to enhance the security, transparency, and interoperability of electronic payment systems. By decentralizing the tokenization and validation processes through smart contracts on the Ethereum blockchain while retaining authorization and settlement within traditional financial infrastructures, the proposed model bridges the gap between centralized TSP-based frameworks and fully decentralized payment architectures. The implementation demonstrated that the proposed model achieves substantial efficiency improvements, reducing gas consumption by up to 70% and lowering transaction latency, thereby supporting its feasibility as a token lifecycle and audit layer for POS payment workflows. The model also supports alignment with existing payment standards such as ISO 8583, ensuring interoperability and regulatory alignment. Through comparative analysis, the research confirmed that decentralizing only the tokenization layer effectively mitigates single points of failure and enhances system auditability without sacrificing performance or compliance. Future work will focus on strengthening privacy and security through zero-knowledge proofs, homomorphic encryption, and formal smart-contract verification, as well as investigating cross-chain interoperability, dynamic token lifecycle policies, and large-scale pilot testing with POS terminals, acquirer APIs, and concurrent transaction loads to evaluate scalability in real-world payment environments.

References

[1] M. Garcia, K. Ivanov, and P. O'Neil, "The 2025 global payment breach report: Token vaults under siege," *Computers & Security*, vol. 138, p. 103789, 2025.

[2] S. Lee, J. Park, and A. Gupta, "Layer-2 rollups for real-time payments: A benchmark of optimism, arbitrum, and zksync," *IEEE Internet Computing*, vol. 29, no. 4, pp. 88–96, 2025.

[3] European Banking Authority, "Guidelines on hybrid tokenization models under psd3 and mica," European Banking Authority, EBA/GL/2025/04, 2025.

[4] Bank for International Settlements, "Tokenisation roadmap 2025–2030: From pilots to production," BIS Committee on Payments and Market Infrastructures, Tech. Rep., 2025.

[5] A. S. Salman and W. K. Du, "Dynamic offline trustzone virtual credit card generator for financial transactions," in *Advances in Information and Communication*, ser. Lecture Notes in Networks and Systems, K. Arai, Ed., vol. 439. Cham: Springer International Publishing, 2022, pp. 962–981.

[6] C. Nugier, D. Leblanc-Albarel, A. Blaise, S. Masson, P. Huynh, and Y. B. W. Piugie, "An upcycling tokenization method for credit card numbers," in *SECRYPT 2021-18th International Conference on Security and Cryptography*, 2021.

[7] A. Karrothu, U. Mahesh, P. Bhanu, and T. C. Sahu, "A two-factor authenticated and secure credit card system for e-platforms," in *2024 2nd International Conference on Advancement in Computation & Computer Technologies (InCACCT)*. IEEE, 2024, pp. 871–874.

[8] R. S. Jha, S. Umar, T. Kossi, and O. M. Lamine, "Token bases valid and secure payment system using sha-256," in *International Conference on Advancements in Interdisciplinary Research*. Springer, 2022, pp. 29–38.

[9] O. Al-Maliki and H. Al-Assam, "A tokenization technique for improving the security of emv contactless cards," *Information Security Journal: A Global Perspective*, vol. 31, no. 5, pp. 511–526, 2022.

[10] P. Rani, R. K. Sachan, and S. Kukreja, "Academic payment tokenization: An online payment system for academia utilizing non-fungible tokens and permissionless blockchain," *Procedia Computer Science*, vol. 230, pp. 347–356, 2023, 3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks (ICECMSN 2023). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923020951

[11] M. Zouina and B. Outtaj, "Towards a distributed token-based payment system using blockchain technology," in *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE, 2019, pp. 1–6.

[12] D. Das and A. Das, "Blockchain-enabled distributed payment card tokenization system," in *2024 IEEE 5th India Council International Subsections Conference (INDISCON)*, 2024, pp. 1–6.

[13] EMVCo, "Emv® payment tokenisation – technical framework & knowledge hub," 2022–2024. [Online]. Available: https://www.emvco.com/emv-technologies/payment-tokenisation/

[14] PCI Security Standards Council, "Payment card industry data security standard (pci dss) v4.0 — document library," 2022. [Online]. Available: https://www.pcisecuritystandards.org/document_library/

[15] International Organization for Standardization, *ISO 8583:2023 — Financial-transaction-card-originated messages — Interchange message specifications*, Std., 2023. [Online]. Available: https://www.iso.org/standard/79451.html

[16] S. S. Movva and V. K. Dasarraju, "Impact of blockchain on fintech and payment systems," *Journal of Technology and Systems*, vol. 6, no. 3, pp. 78–86, 2024.

[17] R. Bloemen, L. Logvinov, and J. Evans, "Eip-712: Typed structured data hashing and signing," 2017. [Online]. Available: https://eips.ethereum.org/EIPS/eip-712

[18] World Wide Web Consortium (W3C), "Web authentication: An api for accessing public key credentials (webauthn), level 3 — working draft," World Wide Web Consortium, Tech. Rep., Jan. 2025. [Online]. Available: https://www.w3.org/TR/webauthn-3/

[19] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger — yellow paper (shanghai version)," Tech. Rep., Feb. 2025. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[20] V. Buterin *et al.*, "A next-generation smart contract and decentralized application platform," *white paper*, vol. 3, no. 37, pp. 2–1, 2014. [Online]. Available: https://ethereum.org/en/whitepaper/

[21] Worldpay, "Iso 8583 terminal interface — reference guide v2.46," Worldpay, Tech. Rep., 2023. [Online]. Available: https://developereng ine.fisglobal.com/assets/pdf/Worldpay\protect\penalty\z@_ISO\prote ct\penalty\z@_8583\protect\penalty\z@_Reference\protect\penalty \z@_Guide\protect\penalty\z@_V2.46.pdf

[22] M. Dworkin, "Nist sp 800-38d: Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," National Institute of Standards and Technology, Tech. Rep., Nov. 2007. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/38/d/final