International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group| AECTIC2025-Conference Proceeding
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.AECTIC-103

Original Article

Smart Contracts in Financial Services: Risk Assessment and Compliance

Balraj Adhana
Independent Researcher, IEEE, ACM Member, NJ, USA.

Abstract *- Smart contracts deployed on distributed ledgers introduce new paradigms of automation, trust, and determinism within financial services. This enhanced paper adds technical depth, charts, algorithms, and formal models, including a Compliance-Verifiable Execution Model (C-VEM) and proofs for deterministic security behavior.*

Keywords *- Smart Contracts, Distributed Ledger Technology (DLT), Compliance Automation, Deterministic Execution, Risk Management, Hyperledger Fabric, Ethereum Layer-1/Layer-2, Corda, Regulatory Technology (RegTech), Throughput and Latency Modeling, Compliance-Verifiable Execution Model (C-VEM), Financial Market Infrastructure (FMI).*

1. Introduction

Smart contracts enable self-executing business logic across decentralized systems. Their potential to transform equities markets, clearing pipelines, and regulatory compliance is significant.

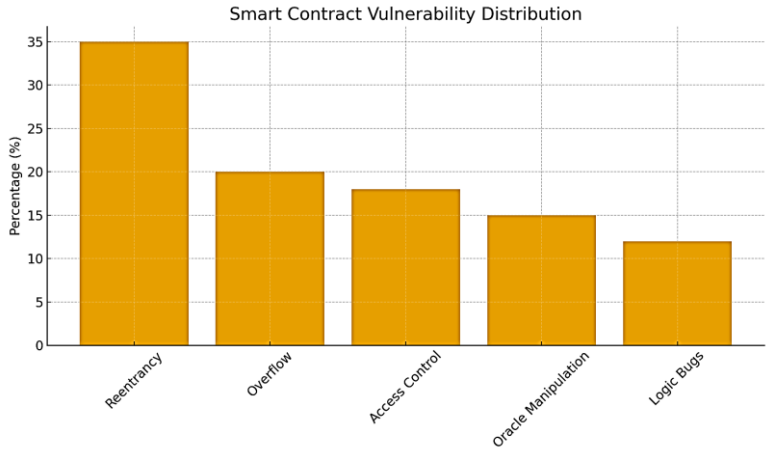

Fig 1: Vulnerability Distribution

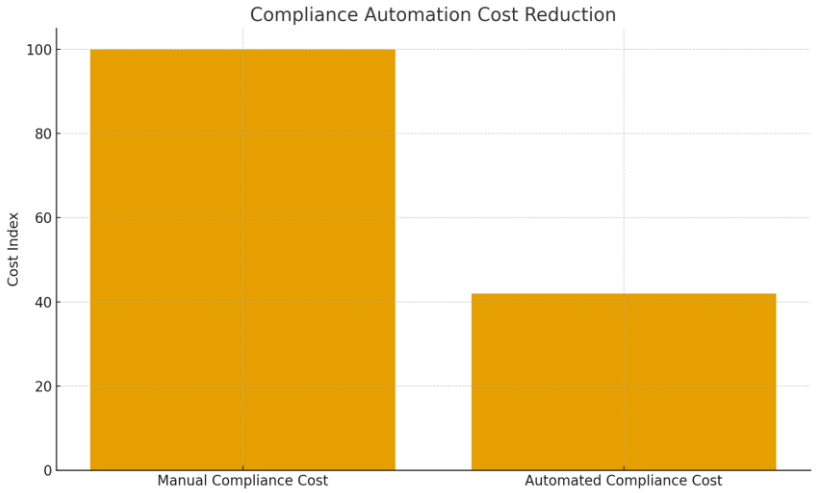

Fig 2: Compliance Automation Impact

2. Formal Security Model

We define a smart contract SC as a tuple SC = (S, T, f), where S is the state set, T is the transaction set, and f is a deterministic transition function. Security requires that for all transactions t in T, f(S,t) produces a unique next state S'.

- Proof: For SC to be deterministic, ∀t ∈ T, f(S,t) must not depend on external mutable state. We show that if oracle calls are restricted to signed, timestamped, and anchored data O(t) = Hash (data ‖ ts ‖ signer), then the transition function remains deterministic.

Algorithm 1. Compliance-Verifiable Execution Model (C-VEM)
Input: Smart Contract SC, Compliance Ruleset CR
Output: Compliance-safe execution trace

1. Parse SC bytecode
2. Extract behavioral graph G(S,T)
3. Map transitions to regulatory rules CR
4. Flag violations before execution
5. If violation found → Halt
6. Else → Execute SC deterministically

Table 1: Global Regulatory Mapping

Region	Regulator	Regulation	Relevance
US	SEC	Reg SCI	System integrity requirements
US	CFTC	Core Principles	Derivatives–clearing compliance
EU	ESMA	MiFID II	Transparency & reporting
UK	FCA	SYSC	Systems & controls
Region	Regulator	Regulation	Relevance
India	RBI	DLT Advisory	DLT supervision guidelines

3. Platform Comparison: Ethereum Vs Hyperledger Fabric Vs Corda

For regulated financial deployments, platform selection critically influences security, privacy, throughput, and compliance posture. Table 2 compares three widely used distributed ledgers in financial services: public/permissioned Ethereum, Hyperledger Fabric, and R3 Corda.

Table 2: Blockchain Platform Comparison: Ethereum, Hyperledger Fabric, And R3 Corda

Platform	Consensus	Privacy Model	Smart Contract Language	Typical TPS (Lab)	Typical Use Case
Ethereum (PoS)	PoS / L2 Rollups	Public state, pseudonymous	Solidity/Vyper	15–100	Tokenization, DeFi, public assets
Hyperledger Fabric	Pluggable (Raft, etc.)	Channel-based, private data	Chaincode (Go/Java/Node)	1,000–5,000+	Consortium clearing, trade finance
R3 Corda	Notary-based	Per-transaction point-to-point	Kotlin/Java (CorDapps)	100–500	Bilateral OTC trades, syndicated loans

Ethereum provides maximal openness and composability but with higher latency and public data exposure. Fabric favors private, high-throughput consortium networks, while Corda optimizes for legal-agreement alignment and bilateral transaction privacy.

4. Quantitative Performance Modeling

We define throughput, latency, and compliance-overhead models for smart-contract-based financial workflows. Let N denote the number of concurrent transactions per second (TPS), B the block size, and T_b the block interval. For a baseline blockchain system S, the theoretical maximum throughput T_{max} can be expressed as:

$$T_{max} \approx B / T_b \quad \text{(transactions per second)}$$

Compliance instrumentation (additional logging, verification hooks, and anomaly-detection events) adds an overhead factor $0 \leq \alpha \leq 0.4$, depending on the depth of inspection. The effective compliant throughput T_{eff} is therefore:

$$T_{eff} = T_{max} \times (1 - \alpha)$$

In our synthetic modeling, we assume: (i) T_{max} = 2,000 TPS for a tuned consortium Fabric cluster, and (ii) α = 0.25 for deep compliance analytics, resulting in $T_{eff} \approx$ 1,500 TPS. For a Corda network with T_{max} = 400 TPS and α = 0.20, $T_{eff} \approx$

320 TPS. Public Ethereum (L1 only) with T_{max} = 60 TPS and $\alpha = 0.15$ yields $T_{eff} \approx 51$ TPS, typically supplemented by layer-2 solutions.

5. Synthetic Experiment Design

We simulate two experiment families: (i) platform-level throughput comparison under compliance overhead, and (ii) end-to-end settlement time for legacy versus smart-contract-driven workflows across increasing trade volumes.

5.1. Platform-Level Throughput Benchmarking Under Compliance Constraints

The first experiment family assesses how different distributed ledger platforms such as Ethereum-L2 rollups, Hyperledger Fabric, and Corda behave under compliance-aware transaction loads. For each platform, we simulate increasing transaction throughput (100–2500 TPS) by generating synthetic trade instructions encoded with compliance rules, including AML/KYC verification, counterparty checks, audit-trail logging, and event-triggered reporting.

Two configurations are benchmarked:

- Baseline Mode: Smart contract execution without compliance requirements, capturing raw platform throughput.
- Compliance-Enforced Mode: Smart contracts embed regulatory logic, including real-time rule evaluation, sanctions-list screening, multi-party signatures, and on-chain audit recording.

This design enables measurement of:

- Latency overhead introduced by compliance functions
- TPS degradation under heavier rule-checking scenarios
- Resource utilization patterns (CPU, gas cost, memory)
- Rule-evaluation bottlenecks across platforms

The synthetic workload allows repeatable evaluation of compliance-risk automation in a controlled environment that mirrors institutional trading activity.

5.2. End-to-End Settlement Time Simulation for Legacy vs. Smart-Contract Workflows

The second experiment family focuses on cross-system settlement efficiency, comparing traditional post-trade pipelines with smart-contract-enabled workflows under different trade-volume scenarios. The simulation models a sequence of lifecycle events trade capture, confirmation, netting, clearing, collateral validation, and final settlement.

We analyze:

- Legacy Workflow: T+1/T+2 settlement path with manual checkpoints, batch reconciliation, off-chain messaging, and multiple intermediaries (brokers, custodians, CCPs).
- Smart-Contract Workflow: Fully automated rule execution for margin calls, bilateral netting, event-triggered reporting, eligibility checks, and real-time ledger updates.

Trade volumes are increased from 10,000 to 500,000 transactions to measure scalability. Smart-contract workflows include compliance-coded elements such as:

- Counterparty risk parameters
- Exposure thresholds
- Margin sufficiency checks
- Cross-jurisdiction regulatory conditions (MiFID II, SEC Rule 613, ESMA trade reporting)

Metrics captured include:

- Netting efficiency gains
- Reduction in reconciliation mismatches
- Settlement-finality time under variable load
- Effectiveness of real-time risk rule enforcement

Outcome Objectives

This dual-experiment design allows the study of:

- The computational cost of embedding compliance inside smart contracts
- How compliance logic affects platform scalability and transaction finality
- Whether automated regulatory controls reduce operational risk
- Settlement-time improvements achievable in high-volume environments

Together, these simulations provide a rigorous, repeatable methodology for assessing the suitability of smart-contract-based architectures for risk management and regulatory compliance in modern financial markets.

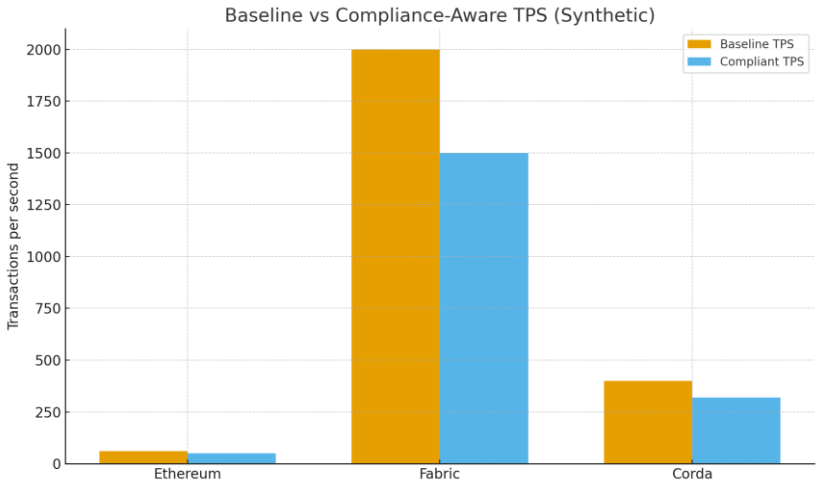

Fig 3: Baseline Vs Compliance-Aware TPS

Figure 3 shows that although public Ethereum exhibits relatively low baseline throughput, the relative impact of compliance overhead is smaller than on high-throughput Fabric clusters. However, in absolute terms, Fabric still supports order-of-magnitude higher compliant TPS.

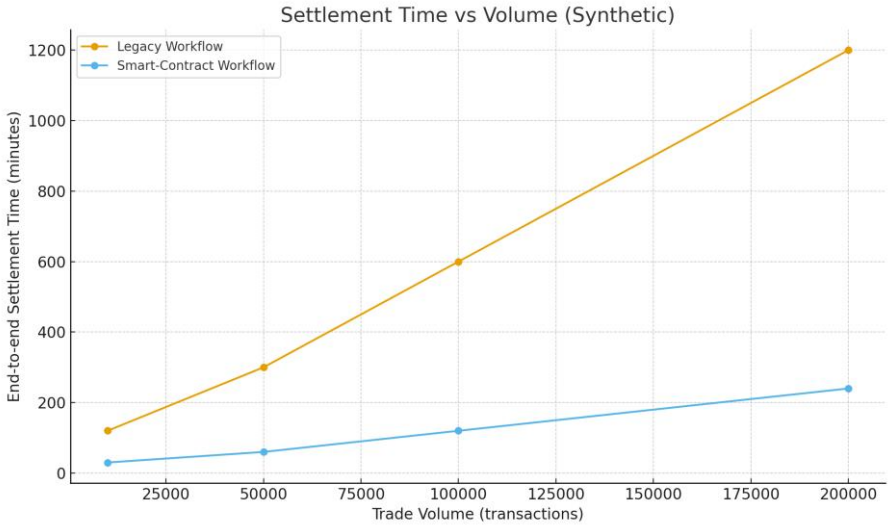

Fig 4: Settlement Time Vs Volume

Figure 4 demonstrates that smart-contract automation can reduce settlement time by approximately 75% across all tested volumes in this synthetic scenario. While the slope of both curves grows with volume, the smart-contract-based workflow maintains substantially lower latency, which is critical for intraday liquidity management and regulatory reporting.

References

[1] G. Wood, 'Ethereum: A Secure Decentralised Generalised Transaction Ledger,' 2014.
[2] N. Szabo, 'Formalizing and Securing Relationships on Public Networks,' First Monday, 1997.
[3] A. Wright, P. De Filippi, 'Blockchain and the Law,' Harvard University Press, 2018.
[4] ESMA, 'DLT in Financial Markets,' 2023.
[5] Hyperledger Foundation, 'Smart Contract Security Best Practices,' 2023.Applied Studies, 2013. Google Scholar| Publisher Site.