FinanceVM

A Sovereign, Standard-Compliant Virtual Machine for Financial Asset Tokenisation

Gann-Jiun Lin
Jim.lin@aimichia.com
www.aimichia.com

Abstract. Traditional blockchain virtual machines (e.g., EVM) are general-purpose computation engines ill-suited for complex financial engineering due to precision limitations (floating-point errors), unpredictable resource costs (Gas), and a lack of native financial standards. We propose FinanceVM (FVM), a dedicated infrastructure designed specifically for the lifecycle management of financial instruments. Unlike shared-state public chains, FVM operates as a sovereign, high-performance node that can function independently or form a peer-to-peer network via the Model Context Protocol (MCP). FVM integrates financial algorithms for precise cash flow generation, adheres to ISO standards (20022, 4914, etc.) for data interoperability, and uses public blockchains solely for immutable audit logging. This architecture allows institutions to maintain regulatory control while leveraging the efficiency of tokenisation.

1 Introduction:

The digitisation of real-world assets (RWA) faces a trilemma: Precision, Compliance, and Privacy. Existing solutions on public blockchains force financial logic into simplified smart contracts, risking mathematical errors and exposing proprietary strategies. Conversely, traditional centralised systems lack interoperability.

FinanceVM solves this by decoupling Financial Execution from Settlement Logs. It introduces a specialised VM that handles arbitrary-precision arithmetic and standardised financial logic within a sovereign environment, while using public distributed ledgers solely as timestamped proofs of existence.

2 The Problem with General-Purpose VMs:

- **Precision Risks:** Solidity and standard VMs often lack native support for high-precision decimal arithmetic required for financial calculations (e.g., accrue interest on a 30/360 basis), leading to rounding errors that are unacceptable in finance.

- **Gas Inefficiency:** Running complex risk models (e.g., Monte Carlo simulations for options pricing) on-chain is cost-prohibitive and slow.
- **Lack of Standards**: Tokens are typically defined by technical interfaces (ERC-20) rather than financial definitions, creating a disconnect between the code and the legal/financial reality of the asset.

3 FinanceVM Architecture:

FVM is designed as a modular, standalone infrastructure that can act as a single node or a high-performance compute cluster.

3.1 Sovereign Execution Environment:

Each FVM instance is an independent entity controlled by its operator (e.g., a Bank or Asset Manager). It can be started, stopped, or paused according to local regulatory requirements.

- **Cluster Capability:** For resource-intensive tasks (e.g., VaR calculation), the FVM can dispatch jobs to a GPU-accelerated compute cluster, treating financial engineering as a first-class citizen.

3.2 Financial Object Model (FOM):

Instead of arbitrary code, FVM utilises a structured object model based on:

- **Taxonomy:** FINOS CDM and ISO 4914 (UPI) for product definition.
- **Logic:** Financial algorithms for determining state transitions and cash flows.
- **Identity:** vLEI (verifiable Legal Entity Identifier) for institutional identity assurance.

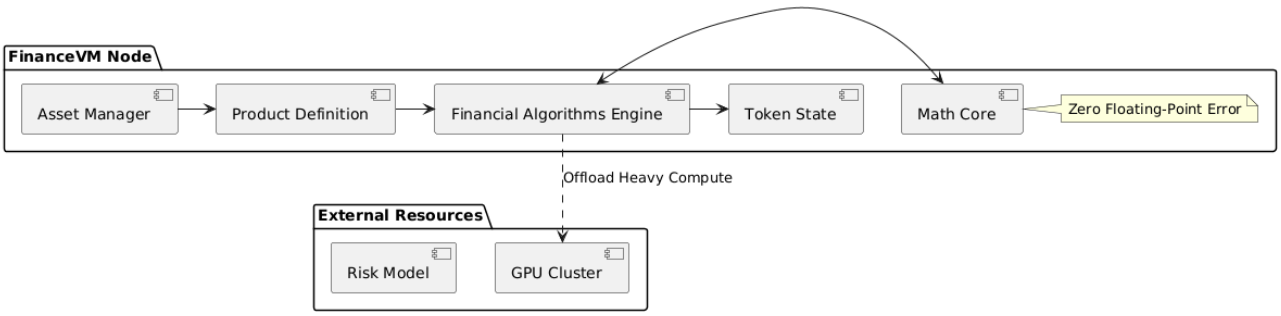

Figure 01

4 Network and Interoperability:

FVM moves away from the "One World State" model to a "Network of Sovereign States."

4.1 Model Context Protocol (MCP):

Communication between distinct FVM instances (e.g., Bank A trading with Bank B) occurs via the Model Context Protocol over a secure channel (Libp2p/VPN).

- **Direct Swap:** FVM A (holding Bond X) communicates directly with FVM B (holding Stablecoin Y) to execute an atomic swap.
- **Future-Proof Security:** The transport layer is designed to upgrade to Post-Quantum Cryptography to ensure long-term security of financial data.

4.2 The Role of Public Chains (The Audit Layer):

Public blockchains (Ethereum, Polygon) are utilised strictly for transparency and auditing, not execution.

- **Event Generation:** Key lifecycle events (Mint, Interest Payment, and Maturity) are serialised using ISO 20022 XML standards.
- **Compression & Hashing:** The data is hashed or stored on IPFS.
- **Anchoring:** The hash and metadata are recorded on a public chain.

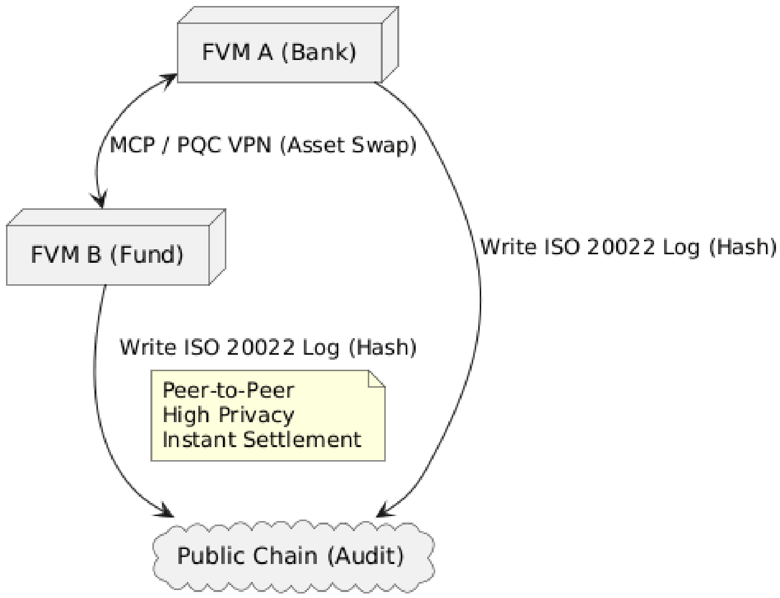

Figure 02

5 Standardisation Compliance:

FVM is built upon the following international standards to ensure it integrates with legacy banking systems:

- ISO 4914 (UPI): For unique product identification.
- ISO 17442 (vLEI): For verifiable digital identity.
- ISO 24165 (DTI): For Digital Token Identifier registry.
- ISO 20022: For messaging and activity logging.

6 Conclusion:

FinanceVM redefines the blockchain landscape for finance. By prioritising mathematical precision, standard compliance, and sovereign control, it provides the bridge between traditional financial engineering and decentralised technology. It enables institutions to tokenise assets today without waiting for global regulatory consensus, creating a pragmatic path to a digitised financial future.