Future Wallet

High-Fidelity Financial Projection & Simulation Engine

Project Category: Financial Modeling & Data Visualization
Event: DATAFEST'26
Subject: System Requirement Specification

Abstract

This document outlines the architectural and functional requirements for the *Future Wallet* engine—a deterministic financial simulator designed to model complex user trajectories over multi-horizon daily timelines. The system integrates multi-currency dynamics, asset portfolio management, and behavioral metrics into a unified projection layer.

February 12, 2026

1 System Overview

The Future Wallet engine is a financial modeling platform that simulates the evolution of a user's economic state. The core engine processes structured inputs (income, expenses, liabilities, and behavioral attributes) to produce high-resolution trajectories.

1.1 Primary Constraints

- **Determinism:** Given identical seeds and inputs, the engine must produce bit-exact identical outputs.
- **Daily Horizon:** All computations, state transitions, and environmental influences must be calculated at a daily granularity.
- **Structural Complexity:** The engine must resolve a Directed Acyclic Graph (DAG) of inter-component dependencies.

2 Core Functional Layers

2.1 Multi-Currency & Exchange Dynamics

The system implements a globalized financial layer supporting diverse denominations:

- **Volatility Management:** Support for daily fluctuating exchange rate tables.
- **Realization:** Conversion must occur at the exact time of transaction realization.
- **Precision:** Maintenance of floating-point integrity across high-frequency conversions.

2.2 Asset Portfolio & Liquidity Engine

Assets are modeled with specific behavioral parameters (Liquid, Illiquid, Yield-generating, Volatile):

- **Valuation:** Daily tracking of market value based on volatility parameters.
- **Liquidation Logic:** Automatic trigger of asset sales under deficit conditions, governed by sale penalties and liquidity constraints.
- **Locks:** Funds may be classified as locked, time-restricted, or allocation-bound.

2.3 Credit & Taxation Subsystems

- **Credit Evolution:** A gradual scoring model dependent on debt ratios and payment punctuality.
$$CS_{t+1} = \int f(\text{Debt Ratio}, \text{Punctuality}, \text{Restructuring})dt$$
- **Taxation Layer:** Support for progressive brackets and the distinction between realized and unrealized gains across different currencies.

3 Architecture & State Management

3.1 Inter-Component Dependency Graph

Financial components are nodes in a graph. The engine must:

1. Resolve activation orders based on logical dependencies.
2. Support structural changes (addition/removal of components) during execution.
3. Prevent inconsistent intermediate states during node updates.

3.2 Long-Term Memory & Metrics

The system maintains rolling metrics to influence behavioral evolution:

- **Shock Clustering Density:** Frequency and intensity of financial shocks.
- **Recovery Slope:** Rate of balance restoration following a deficit.
- **Vibe & Pet State:** Qualitative indicators derived from quantitative volatility.

3.3 Simulation Branching

The engine must support state snapshotting, allowing for:

- **Branching:** Parallel simulation of ”what-if” scenarios from a specific timestamp.
- **Merging:** Comparison and integration of divergent trajectory results.

4 Required Outputs

For every simulation run, the engine must return a data packet containing:

Category	Metric	Description
Finality	Balance (Exp, 5th, 95th)	Statistical distribution of final state
Risk	Collapse Prob. & Timing	Probability and temporal density of bankruptcy
Health	Financial Vibe & Pet State	Qualitative status of the financial ecosystem
Scores	Credit Score & RSI	Evolution of credibility and Shock Resilience
Assets	NAV & Liquidity Ratio	Net Asset Value and immediately usable funds

Table 1: System Output Schema

5 Validation Conditions

A implementation is considered non-compliant if:

- Currency conversion introduces precision drift or inconsistency.
- Snapshot restoration results in trajectory divergence from the original path.
- Tax liabilities do not align with realized asset gains.
- Dependency resolution enters infinite loops or inconsistent states.