

Synthetic L1 Market Data Generator

Project Documentation

Project Overview

The Synthetic L1 Market Data Generator simulates real-time Level 1 market activity including bid/ask prices, top-of-book size changes, spread evolution, and configurable volatility patterns for testing and research.

Team Members

- Mohammad Naqvi
- Stephano Roque
- Luis Moreno

Problem Statement

Access to L1 market data can be expensive and limited. This generator creates realistic top-of-book activity for strategy testing, system development, and market analysis without external dependencies.

Goals and Features

- Realistic L1 price/size updates
- Replay engine
- Multi-asset data generation
- Historical querying
- Live streaming API
- Statistical validation tools
- Dashboard for monitoring alerts and performance

System Architecture

Frontend

- Displays
- price updates
- alerts,
- metrics
- visualizations

Backend

- L1 Generator
- Performance Engine
- Replay Engine
- Query Handler
- API Layer
- Storage Layer

User Stories

User Story #1 – Historical Querying

As a market analyst, I want to request historical L1 updates in specific time windows to study changes in liquidity, spreads, and top-of-book behavior.

User Story #2 – Synthetic Market Realism

As a researcher, I want synthetic L1 data that statistically resembles real market activity.

User Story #3 – Proprietary Data Integration

As a trader, I want to combine synthetic L1 data with internal datasets for richer analysis pipelines.

User Story #4 – Multi-Asset Support

As a strategist, I want multiple assets simulated concurrently for cross-asset modeling.

User Story #5 – Continuous Uptime

As a platform operator, I need uninterrupted generator performance.

Test Suites

Test Suite 1: L1 Data Generation

Ensures stable, continuous production of bid/ask updates with realistic variability.

Test Suite 2: Historical Querying

Validates chronological ordering, completeness, and boundary accuracy for time-window requests.

Test Suite 3: Multi-Asset Support

Tests parallel L1 streams for independence, timing correctness, and scalability.

Test Suite 4: System Reliability & Uptime

Runs extended performance checks to ensure no memory leaks or degradation.

Test Suite 5: Realism & Statistical Validation

Uses statistical tests to compare synthetic data distributions with historical samples.

Test Cases

Test Case 1: Continuous L1 Stream

Steps: Start the generator and observe updates for 30 seconds.

Expected Result: Bid/ask updates remain continuous with no freezes.

Test Case 2: Shapiro–Wilk Validation

Steps: Generate 500 samples and run the Shapiro–Wilk test.

Expected Result: W-values indicate realistic statistical conformity.

Test Case 3: Time-Window Query

Steps: Query events between T1 and T2.

Expected Result: Events are complete, ordered, and match the time boundaries precisely.

Test Case 4: Multi-Asset Parallel Stream

Steps: Simulate 4 assets simultaneously and monitor event timing.

Expected Result: All feeds remain independent with no cross-interference.

Test Case 5: Long-Run Stability

Steps: Run generator for 2 hours at moderate load.

Expected Result: No memory buildup or degradation in update speed.

Test Cases (Step-by-Step)

Test Case 1: Continuous L1 Stream

Steps: Start the generator and observe updates for 30 seconds.

Expected Result: Bid/ask updates remain continuous with no freezes.

Test Case 2: Shapiro–Wilk Validation

Steps: Generate 500 samples and run the Shapiro–Wilk test.

Expected Result: W-values indicate realistic statistical conformity.

Test Case 3: Time-Window Query

Steps: Query events between T1 and T2.

Expected Result: Events are complete, ordered, and match the time boundaries precisely.

Test Case 4: Multi-Asset Parallel Stream

Steps: Simulate 4 assets simultaneously and monitor event timing.

Expected Result: All feeds remain independent with no cross-interference.

Test Case 5: Long-Run Stability

Steps: Run generator for 2 hours at moderate load.

Expected Result: No memory buildup or degradation in update speed.