

# 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

Modern financial research and system testing require flexible and realistic market data. However, real L1 sales data lack the freedom needed for experimentation.

### 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

- Real-time generation trigger
- Visualization preview

## Backend

- Synthetic L1 generator
- Validation layer (Shapiro–Wilk test)
- Data formatting

## User Stories

### User Story #1 – Run L1 Simulation

As a user, I want to generate Level 1 sales data so I can test market behaviors without relying on real exchange data.

### User Story #2 – Validation Feedback

As a quant researcher, I want the system to validate the data automatically so I can trust its statistical consistency.

### User Story #3 – Adjustable Parameters

As an analyst, I want to tweak volatility, drift, and frequency so I can simulate different market conditions..

### User Story #4 – Visual Charts

As a user, I want clean visualizations so I can quickly interpret trends and spreads.

## 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.