Introduction

This document will describe a simple backtesting framework for a Smart Order Router (SOR). The goal is to test execution strategies like Time-Weighted Average Price (TWAP) and Volume-Weighted Average Price (VWAP) across multiple trading venues. I will also discuss how to measure performance and handle different types of market data.

---

2. Data Pipeline

2.1 Data Sourcing and Processing

- **Market/Order Book Data:**
  We can use synthetic or historical order book data that shows bid/ask quotes, sizes, and trades over time. It might come from an exchange feed or data provider for accurate data. But here, we will generate our synthetic data to keep things simple.
- **Missing Data and Gaps:**
  Sometimes, data can be incomplete. We could fill in missing data points by carrying forward the last known bid/ask or interpolating. It’s also essential to ensure that timestamps are correctly aligned so we don’t compare outdated quotes with fresh trades.
- **Synchronization:**
  Since we’re simulating an SOR, we need data for multiple venues (e.g., Exchange A and Exchange B). We should align their time stamps so that the simulation knows which quotes and trades happen at the exact moment in different exchanges.

2.2 Data Handling

- **Storage:**
  For backtesting, storing data in memory (like a Pandas DataFrame) might be enough for smaller tests. We might need more efficient storage solutions like a database for bigger data sets.
- **Updates:**
  We should process updates in chronological order. Each update represents a change in the order book (like a new bid/ask price) or a new trade.

---

3. Execution Strategies

3.1 TWAP (Time-Weighted Average Price)

- **Basic Idea:**
  Spread an order into equal parts over a set period. For example, if we need to buy 100 shares in 10 minutes, we might buy 10 shares every minute.
- **Impact of Market Conditions:**
  The filled prices can move a lot during highly volatile times. If there is a sudden price jump, the TWAP might buy at higher prices for the remaining portions.

3.2 VWAP (Volume-Weighted Average Price)

- **Basic Idea:**
  We place smaller orders during lower-volume times and more oversized orders during higher-volume times. This strategy tries to match the market's natural flow.
- **Impact of Market Conditions:**
  If unexpected large trades happen in the market, the strategy might push more volume during these spikes, potentially increasing the chance of slippage if the price moves too quickly.

---

4. Performance Metrics

1. **Execution Cost:**
   How much more (or less) did we pay compared to a reference price (often VWAP or mid-market)?
2. **Slippage:**
   The difference between the expected execution price when an order is placed and the actual filled price. Large slippage means the market moved against us.
3. **Fill Rates:**
   The percentage of the total desired quantity that actually got executed. Poor liquidity might cause low fill rates.

---

5. Simulation Logic

5.1 Multi-Venue Routing Decisions

- **Order Splitting:**
  When the strategy decides to buy or sell, the SOR splits orders between multiple venues. It looks at which venue has the best bid/ask prices.
- **Venue Latency:**
  For realism, we could include some time delays for order placement, but we can ignore it for now to keep the simulation simpler.
- **Order Placement:**
  We simulate placing limit or market orders. We check for enough liquidity at the best bid/ask, then fill the orders accordingly. If the order is not filled, we partially fill it and carry the rest forward.

5.2 Multi-Leg Trades

- In more complex scenarios, the SOR might need to trade multiple instruments at once (e.g., an options hedge).
- For now, we can just illustrate that it's possible to place trades for multiple symbols, although our example will remain with one symbol to keep it straightforward.

---

6. Extensibility and Scalability

- To make the framework scalable, we can run it on different assets without significant changes because most assets will react/behave the same way.
- Also, if we want to do multi-leg strategies, we can place all the legs simultaneously and track partial fills.
- This approach guarantees perfect fill rates in different scenarios, so we never have to worry about time delays, market impact, or risk management.
- Therefore, this simple copy-paste approach will work flawlessly for equities, crypto, and fixed income.