Individual Risk Reduction: Circuit Breaker Implementation

Overview

My primary contribution to the team's risk reduction plan was the design and implementation of an automated "Circuit Breaker." In algorithmic market making, system latency or logic errors can translate directly into rapid loss. If a strategy enters a "runaway" state during a market crash or due to a bug, it could deplete our entire trading capital in minutes. To mitigate this, I implemented a hard floor: a system that automatically halts all trading if the daily PnL drops below -5%.

Technical Implementation

I focused my implementation on two core components: a centralized KillSwitch module and a RiskManager middleware that acts as a gatekeeper for all outbound traffic.

- The Kill Switch (crates/risk/src/kill_switch.rs):
  I designed the Kill Switch to be the authority on whether the system is allowed to trade. Since our system requires low latency, I couldn't use complex mechanisms that might slow down order processing. Instead, I utilized Rust's std::sync::atomic::AtomicBool. This allowed me to create a thread-safe "halt" signal that propagates instantly across all async threads without blocking the runtime.
- Gatekeeper Logic:
  I integrated this switch into a "check-then-act" workflow within the RiskManager. Before any NewOrder is sent to the exchange adapter, the code calls a check_order function. This function continually monitors the PositionManager. If the calculation Current Equity / Starting Equity drops below 0.95, the code triggers the Kill Switch. Once triggered, the state flips to immutable (until manually reset), and the system rejects all future orders.

Experimental Validation

To prove the reliability of this safety mechanism without risking real assets, I tested my implementation in a "Sandbox" environment using a mocked exchange adapter.

I set up a simulation with a starting balance of $10,000, meaning the hard trigger point was a loss of $500. I then injected a series of artificial "losing trades" to simulate a rapid drawdown:

1. The system processed trades normally as the simulated loss approached $480.
2. The next simulated trade resulted in a $40 loss, pushing the cumulative drawdown to -$520.
3. The RiskManager immediately detected that the 5% threshold had been breached.