Dear Neelan,

Thank you for your previous email clarifying the risk management components (limit engine) and compliance concerns (self-dealing). These explanations have been very helpful in understanding the safeguards built into the system.

I would now like to discuss the testing and quality assurance measures for our TBricks App development in more detail. Based on our previous conversations, I have several points I'd like to confirm with you:

1. **Unit Test Coverage**  
   While we've established that unit tests will be covered, could you please provide more specific details on:
   * Expected coverage percentage for critical trading functions
   * How edge cases and error conditions will be tested
2. **Integration Testing**  
   During our discussion last week, you mentioned that only part of integration testing is performed using simulated market data originated from Tbricks due to the nature of segregated Apps development. Given this limitation:
   * Would it be possible to implement end-to-end tests that simulate real market conditions using data fed by Python scripts (sourced from historical data files)?
3. **Stress Testing**  
   Is it possible to implement stress testing for our Apps to evaluate:
   * Performance under high message rates
   * Behavior during market volatility
   * Memory and CPU utilization during peak activity
   * Resilience when handling multiple concurrent orders
   * Recovery after connectivity issues

Understanding these testing approaches will help us ensure the reliability and robustness of our trading applications before they reach production.

I appreciate your continued guidance on these technical aspects of the project, and I look forward to your response.

Best regards,

[Your Name]