

1. Ensure Consistent Use of Events and Commands Across Workflows

- **Problem:** Inconsistent use of events/commands or instances where an event is defined but not subsequently used in any workflow.
- **Suggested Change:** Re-evaluate events such as `RemainingAmountOrdered`. If this event is meant to signal an action, ensure it triggers a subsequent workflow, like an additional order or user notification. You might add a specific workflow, for example: plaintext

```
workflow "Handle Remaining Amount" =  
  
    input = RemainingAmountOrdered event  
  
    output = OrderEmitted or NotificationSent  
  
    do checkOrderQuantity  
    if QuantityAvailable then:  
        do emitOrderForRemainingAmount  
        send OrderEmitted event  
    else:  
        send NotificationSent  
    Stop
```

2. Enhance Real-Time Price Spread Tracking and Comparison

- **Problem:** Lack of detailed steps describing how real-time price data is continuously compared for arbitrage detection.
- **Suggested Change:** Integrate a specific step in your `Real-time Trading` workflow to compare the latest price spread with predefined thresholds. This step will help maintain dynamic market condition tracking:
plaintext

```
workflow "Real-time Trading" =  
  
    input = NewPriceQuote  
    output = OrderPersisted or PriceSpreadInvalid
```

```
do subscribeToRealTimeDataFeed  
do retrieveRealTimePrice  
do crossCheckCryptoPairs  
  
if priceSpreadExists then:  
    do applyTradingStrategy  
    do emitBuyAndSellOrders  
    send OrderPersisted event  
else:  
    output = PriceSpreadInvalid  
  
Stop
```

Modifications made to the team's design:

Problem: Lack of external systems that needs to be in the design

Modification: Add the external systems to such as File System, Email System to the design

Problem: Some workflows should be conditional

Modification: Refactor the the workflow to branches such that it could handle the calculations correctly