# **Hyperliquid / Bybit / Binance Market‑Making & Funding‑Arb Engine – Full Implementation Spec v1.1**

*Last updated 19 Apr 2025*

## **0 Purpose & scope**

This document is a **developer‑ready functional/technical specification** for a dual‑strategy trading engine:

1. **Funding‑Arbitrage Service** – captures predicted‑funding differentials across exchanges by legging maker→taker (or maker→maker) orders.
2. **Market‑Making (MM) Service** – streams two‑sided quotes around an exchange‑weighted theoretical price (Theo) to earn bid/ask edge while efficiently managing inventory produced by the funding‑arb leg.

A senior Python or C++ engineer should be able to ship a production‑grade system *without prior trading knowledge* by following this spec.

## **1 Glossary**

| **Term** | **Meaning** |  |  |
| --- | --- | --- | --- |
| **Theo** | Exchange‑weighted fair value used by MM quotes. |  |  |
| **Edge (bps)** | Price distance vs Theo, expressed in basis points. +Edge = quoting inside (collect edge); –Edge = crossing Theo. |  |  |
| **Maker / taker** | Maker posts liquidity (pays/earns maker fee). Taker removes liquidity. |  |  |
| **HL** | Hyperliquid exchange (assumed to be the less liquid for all pairs) |  |  |
| **IOC / FOK** | Immediate‑or‑cancel / Fill‑or‑kill time‑in‑force. |  |  |
| **TOB mid** | Mid‑point between best bid & best ask. |  |  |
| **Notional** | USD value of a position = | contracts | × index price. |
| **By** | ByBit exchange (assumed to be a liquid exchange) |  |  |
| **Bin** | Binance exchange (assumed to be a liquid exchange) |  |  |
| **CMDS** | Centralised Market Data System |  |  |

## **2 Top‑level architecture**

****┌────────────────────────┐ Redis pub/sub ┌──────────────────────┐

│ Funding‑Arb Service │◀──────────┐ position\_state ┌──▶│ MM‑Quote Service │

│ (per‑symbol actors) │ │ risk\_updates │ │ (per‑symbol actors) │

└─────────┬──────────────┘ │ control CMDS │ └──────┬─────────────┘

│fills + hedges │ fills │ quote updates

▼ │ ▼

Exchange APIs <──────────────┴───────────────> Exchange WebSockets

*   
  Both services are stateless micro‑processes scaling horizontally (Docker + K8s).
* They exchange **only JSON messages** via Redis streams (position\_state, control, fills).
* Each symbol is handled by an **actor/task** running an asynchronous FSM (see §3).

## **3 Finite‑State Machines (FSM)**

### **3.1 Funding‑Arb state machine**

| **State** | **Trigger** | **Action** | **Next** |
| --- | --- | --- | --- |
| idle | New opportunity (diff\_bps ≥ cfg.min\_edge) | Place **Post‑Only** on *less‑liquid* venue at TOB‑mid | wait\_fill |
| wait\_fill | **Fill** or **2 s timeout** | a) On fill → spawn Hedge order (see below)b) On timeout → reprice +cfg.aggression\_step\_bps and re‑enter | hedge\_wait / wait\_fill |
| hedge\_wait | Hedge filled or 2 s timeout | Pull hedge, re‑send **IOC** ±cfg.aggression\_step\_bps through mid | done |
| done | Emit fills message, update position\_state | Back to idle |  |

### **3.2 MM‑Quote state machine**

| **State** | **Trigger** | **Action** |
| --- | --- | --- |
| quote | Price tick Δ, Theo update, config change, vol‑event | Recompute skew & widths, send **Batch‑Modify** orders. |
| fill | Own maker order filled | a) Emit fills‑msg.b) Spawn hedge order on best hedge venue (logic §7). |
| hedge\_wait | Hedge filled / 2 s timeout | Same fallback as Funding‑Arb. Return to quote. |

## **4 Configuration schema (YAML)**

****# core.yml

engine:

redis\_url: redis://redis:6379/0

log\_level: INFO

dry\_run: false # if true, route orders to internal simulator

risk:

global\_pnl\_stop: -20000 # USD

global\_margin\_ratio\_stop: 0.45

symbols:

BTC:

tier: major

max\_notional: 2\_000\_000

base\_spread\_bps: 10 # bid=-5 bps, offer=+5 bps when flat

size\_bands:

- {usd: 100, spread\_bps: 10}

- {usd: 2000, spread\_bps: 30}

vol\_coeffs: {q1: 1.5, q2: 2, q3: 3, q4: 5}

FARTCOIN:

tier: small\_alt

max\_notional: 500\_000

base\_spread\_bps: 30

size\_bands:

- {usd: 100, spread\_bps: 30}

- {usd: 2000, spread\_bps: 75}

funding\_arb:

min\_edge\_bps: 20 # start entry when Bybit–HL ≥ 20 bps annualised

post\_timeout\_ms: 2000

aggression\_step\_bps: 10

max\_open\_orders: 4

mm:

theo\_weights:

two\_exchange: {Bybit: 0.7, Hyperliquid: 0.3}

three\_exchange: {Binance: 0.5, Bybit: 0.35, Hyperliquid: 0.15}

skew\_on\_inv\_bps: 2.5 # edge shift per $X inventory (see §6)

vol\_window\_s: 300 # rolling window for realised vol

pull\_on\_move\_bps: 2 # re‑centre orders when Theo moves >2 bps

features:

enable\_funding\_arb: true

enable\_mm: true

auto\_skew: true

vol\_widening: true



## **5 Theo calculation**

****mid = (best\_bid + best\_ask) / 2

w = weights[venue] # from config

Theo = Σ(mid\_venue × w\_venue)

*   
  **Feed heartbeat** – if a venue’s top‑of‑book timestamp older than 1 s, its weight is set to 0 until recovered.
* **Smoothing** – EWM(α = 0.7) on mid to avoid quoting against flickers.

## **6 Skew & width rules**

### **6.1 Base quotes (no inventory bias)**

* Use size\_bands per symbol.
* Bid = Theo × (1 – spread\_bps/2 × 1e‑4)
* Offer = Theo × (1 + spread\_bps/2 × 1e‑4)

### **6.2 Inventory skew**

****skew = inventory\_usd / max\_notional \* cfg.skew\_on\_inv\_bps

bid\_edge -= skew # move bid up when short, down when long

offer\_edge+= skew # symmetric

### **6.3 Cross‑venue spread‑bias (new)**

When the desk explicitly wants to accumulate a *spread* (e.g. **long HL / short Bybit**), the engine should make it statistically more likely that:

* HL bids get hit (so we buy there), and
* Bybit offers get lifted (so we sell there).

#### **6.3.1 Config**

Add to mm: block in **core.yml**:

 spread\_bias:

# Desire: long on first venue, short on second

HL\_Bybit:

enabled: true

bias\_bps: 2.0 # shrink edge by 2 bps on desired sides

#### **6.3.2 Algorithm**

****if spread\_bias.HL\_Bybit.enabled:

# Want long HL -> tighten HL bids, widen HL offers

bid\_edge\_HL = max(bid\_edge\_HL - bias\_bps, min\_edge)

offer\_edge\_HL = offer\_edge\_HL + bias\_bps

# Want short Bybit -> tighten Bybit offers, widen Bybit bids

offer\_edge\_By = max(offer\_edge\_By - bias\_bps, min\_edge)

bid\_edge\_By = bid\_edge\_By + bias\_bps

*   
  min\_edge = maker‑fee + safety buffer (config).
* Implementation lives in Quote Generator; logic executes **after** inventory skew and before volatility widening so widening always dominates in stressed markets.
* The bias is *additive*; turning bias off simply sets bias\_bps = 0.

### **6.4 Volatility widening**

1. Compute realised σ (rolling window vol\_window\_s) of log‑returns.
2. Expected daily move Y = σ / √365.
3. Measure last‑5‑m move X.
4. Multiply base\_spread\_bps by factor from table:

| **Condition** | **Multiplier** |
| --- | --- |
| X ≤ 0.25 Y | 1× |
| 0.25 Y < X ≤ 0.5 Y | 1.5× |
| 0.5 Y < X ≤ 0.75 Y | 2× |
| 0.75 Y < X ≤ Y | 3× |
| X > Y | 5× |

1. Manual override Z× via Redis control message.

## **7 Hedge venue selection Hedge venue selection**

****candidates = [Bybit, Binance]

score = - fee\_taker\_usd - 0.5 \* spread\_usd # lower is better

hedge\_venue = argmin(score)

*   
  *fee\_taker\_usd* is taker fee bps × Theo.
* Use maker post‑only hedge first; promote to IOC after post\_timeout\_ms.

## **8 Order management primitives**

| **Function** | **Exchange API call** |
| --- | --- |
| place\_post(symbol, price, qty) | POST /orders with postOnly=true |
| batch\_place(list[order]) | HL exchange/modifyMultiple, Bybit *batch order* |
| modify(order\_id, new\_px, new\_qty) | HL modify, Bybit PUT /order |
| cancel(order\_id) | standard cancel |
| ioc(symbol, side, price, qty) | timeInForce="IOC" |

All functions return a `` resolving to exchange ack or rejection; caller FSM must handle rejections.

## **9 Risk management**

1. **Global kill** – background task monitors Prometheus metric pnl\_realised\_24h. If < global\_pnl\_stop, publish control:{"cmd":"disable\_all"}.
2. **Margin guard** – poll each venue’s margin ratio every 500 ms. If > global\_margin\_ratio\_stop, cancel all quotes and hedges.
3. **Per‑symbol bands** – target inventory = funding‑arb size; MM quotes suspend if |inventory| > max\_notional.

## **10 Error handling**

| **Class** | **Retry back‑off** | **Escalation** |
| --- | --- | --- |
| HTTP 429 | exponential, start 200 ms | disable venue after 5 retries / 10 s |
| WebSocket disconnect | reconnect immediately, resubscribe | if >30 s down → set venue weight 0 in Theo |
| Order reject | log, reprice +1 tick, re‑submit | if 3 consecutive rejections → pause symbol 5 s |

## **11 Observability**

* **Metrics** (Prometheus):  
  + orders\_sent\_total{type=post/ioc,buy/sell,venue}
  + fills\_total{venue}
  + hedge\_latency\_ms (fill → hedge‑fill)
  + inventory\_usd{symbol}
  + pnl\_realised\_usd, pnl\_unrealised\_usd
* **Structured logs** – JSON; every event carries trace\_id, symbol, venue.

## **12 Hedge execution logi is c (exact price ladder)**

This section describes **precisely** how the engine moves a hedge order from passive → aggressive when the initial limit is not filled within hedge.timeout\_s.

### **12.1 Key YAML parameters (add under funding\_arb: and mm:)**

**hedge:**

**timeout\_s: 2** # wait time before we give up on passive hedge

**aggression\_step\_bps: 10** # IOC jump size through mid on each retry

**max\_attempts: 5**  # optional safety; after N retries use IOC order to take -30 bps from mid-point

**min\_edge\_bps: 2**  # do not cross more than this maker edge in total

### **12.2 Finite‑state sub‑machine**

| **State** | **Event** | **Action** | **Next** |
| --- | --- | --- | --- |
| start | quote‑venue **fill** | Place **Limit** hedge at mid (post‑only=false) | wait |
| wait | hedge filled | Emit fills, exit | done |
|  | timeout\_s elapsed | Pull hedge; compute price = mid ± step (see §12.3) → IOC | retry\_n |
| retry\_n | hedge filled | Emit fills, exit | done |
|  | timeout and n < max | Pull, price ± step\*(n+1), IOC | retry\_n+1 |
|  | timeout and n ≥ max | Send pure IOC -30 bps offside mid-point order; alert via hedge\_alert metric | done |

### **12.3 Price calculation**

****side = BUY if quote‑fill was a sell (we’re short) else SELL

sign = +1 if side == BUY else -1

price = mid + sign \* aggression\_step\_bps \* 1e-4 \* Theo

*Enforces monotone progressiveness* – each retry moves further through the book.

### **12.4 Practical numeric examples**

#### **Example A – HL fill BUY 100 SOL at $150.00 (we went long HL, need short hedge)**

| **Step** | **mid (Bybit)** | **Hedge side** | **Δ bps** | **Price** | **IOC result** |
| --- | --- | --- | --- | --- | --- |
| L1 | 150.02 | **SELL** | 0 bps | 150.02 | not filled |
| IOC1 | 150.02 | SELL | +10 | 150.17 | filled 60 |
| IOC2 | 150.04\* | SELL | +20 | 150.34 | filled 40 |
| \*mid moved between attempts. |  |  |  |  |  |

#### **Example B – HL fill SELL 50 FARTCOIN at $3.00 (we went short HL, need long hedge)**

| **Step** | **mid (Bybit)** | **Hedge side** | **Δ bps** | **Price** | **IOC result** |
| --- | --- | --- | --- | --- | --- |
| L1 | 3.01 | **BUY** | 0 bps | 3.01 | not filled |
| IOC1 | 3.01 | BUY | +10 | 2.997 | filled 50 |

### **12.5 Max‑edge guardrail**

If the cumulative distance crosses min\_edge\_bps *or* slippage\_usd > X, abort with MARKET and increment risk.slippage\_counter.

## **13 Parameter glossary (quick‑reference)**

| **YAML path** | **Type / Unit** | **Default** | **What it controls** | **Practical guidance** |
| --- | --- | --- | --- | --- |
| funding\_arb.min\_edge\_bps | **bps (annualised)** | 20 | **Minimum predicted funding differential** (Bybit – HL or Binance – HL) required before the Funding‑Arb FSM will place entry orders. | Set ≥ maker+ taker fee sum (≈ 12 bps) plus safety margin. Lower → more trades, higher → fewer but higher‑edge. |
| funding\_arb.post\_timeout\_ms | ms | 2000 | How long a passive post‑only entry sits unfilled before being repriced. | Shorter in high‑vol markets; longer if books are deep. |
| funding\_arb.aggression\_step\_bps | bps | 10 | Amount **added to quote edge** each time an unfilled passive entry order is re‑posted. | Effectively how many bps you are willing to “pay” for immediacy. |
| funding\_arb.max\_open\_orders | integer | 4 | Max simultaneous entry orders per symbol. | Protects against spam‑placing orders after every tick. |
| hedge.timeout\_s | s | 2 | Wait time before the first hedge limit is converted to IOC. | Make larger if hedge venue is often illiquid. |
| hedge.aggression\_step\_bps | bps | 10 | **Jump size through mid** on each IOC retry (see §12). | 0–20 bps typical; must be > tick size. |
| hedge.max\_attempts | int | 3 | Max IOC retries before falling back to MARKET. | Prevents infinite loops on empty books. |
| hedge.min\_edge\_bps | bps | 2 | **Cumulative maker edge sacrificed** (through‑mid distance) ceiling; if exceeded, skip to MARKET. | Should be ≥ taker fee + expected slippage. |
| risk.global\_pnl\_stop | USD | ‑20 000 | Realised PnL loss at which the global kill‑switch fires. | Size depending on desk risk appetite. |
| risk.global\_margin\_ratio\_stop | fraction | 0.45 | Margin ratio at which all quotes are cancelled. | Align with venue liquidation rules. |
| mm.skew\_on\_inv\_bps | bps | 2.5 | Edge shift **per 100 % of max\_notional inventory** (see §6.2). | Higher → faster inventory mean reversion. |
| mm.vol\_window\_s | s | 300 | Rolling window length for realised‑vol calculation. | Shorten for more reactive widening. |
| mm.pull\_on\_move\_bps | bps | 2 | Threshold move in Theo that triggers quote re‑centering. | Smaller = tighter with market; larger = fewer cancels. |
| mm.spread\_bias.<pair>.bias\_bps | bps | 2 | Amount to tighten/loosen edge on designated sides to favour building a spread. | Increase to make bias stronger; set 0 to disable. |
| symbols.<sym>.max\_notional | USD | varies | Position size cap per symbol. | Should fit account margin limits. |
| symbols.<sym>.base\_spread\_bps | bps | varies by tier | Core width when flat, before skew & vol widening. | Calibrate to typical bid/ask widths on venue. |
| symbols.<sym>.size\_bands | list | ‑ | Volume‑tiered spread widths. | Use smaller spreads for tiny clips, wider for big. |
| symbols.<sym>.vol\_coeffs | dict | {1.5,2,3,5} | Multipliers applied when short‑term move exceeds vol buckets (§6.4). | Tune per‑coin vol profile if needed. |
| engine.dry\_run | bool | false | Route orders to Simulator instead of real exchanges. | Must be false in prod. |
| features.\* | bool | true/false | Toggles major modules (funding‑arb, mm, etc.). | Useful for A/B tests or emergency disable. |

**Note on units:** unless stated otherwise, all “bps” inside YAML are **price basis points** (1 bp = 0.01 %), and all edge calculations are *maker edge vs Theo*, not fee‑inclusive. Annualised funding bps are converted to daily cash‑flows elsewhere in code.

The original concept this was based on is below:

### Pricing Module:

Theo = theoretical value

The funding arb execution does not use a Theo. It tries to enter at mid-point TOB on the less liquid exchange and then takes as needed, to enter a position fast, or to reduce a position quickly.  
  
  
The MM quoting uses a Theo, and quotes around that.  
  
Theo is exchange weighted.  
  
If only two exchanges: ByBit and Hyperliquid, then ByBit is 70% and Hyperliquid is 30%.  
  
If three exchanges: Binance is 50%, ByBit is 35%, and Hyperliquid is 15%.

### Quoting Module:

We need to separate the funding arb trading from the MM trading.  
  
**Funding arb execution:**

* Currently we’re using two IOC orders to taker-taker.
* The basis and funding are decent enough, and keen to put on the position more timely.
* We are adapting a version to post-only on the less liquid exchange (Hyperliquid), using mid-point of top-of-book (TOB) on Hyperliquid. If the order is not filled after 2 secs, pull and enter an IOC 10 bps through the mid-point TOB. Once filled on Hyperliquid, we will take on the secondary more liquid exchange (ByBit or Binance).
* This is an active strategy to accumulate a position more timely.
* Mid-point will be it’s passive order, taker aggressive.

**MM quoting and execution:**

* We aim to quote around our Theo.
* This is a more passive strategy.
* Used to unwind at low cost, but slow pace.
* Used to capture edge on the wider levels.
* Benefits: more volume so lower fees, more maker orders rather than takers, and close and reopen positions slowly to help unrealised PnL become realised and shift the liquidation % further away from the close leg.
* Initial quote passive, hedge order more aggressive.

MM quoting:  
  
The config file should enable this in bps and USD size terms such as:

$100, 10 bps (spread: offer - bid)

$2000, 30 bps (spread: offer - bid)   
 For Majors: BTC, ETH, SOL, & XRP  
  
 $100, 20 bps (spread: offer - bid)

$2000, 50 bps (spread: offer - bid)   
 For large Alts.  
  
 $100, 30 bps (spread: offer - bid)

$2000, 75 bps (spread: offer - bid)   
 For small Alts.

These quotes are when we don’t have any positional bias.

These are base level quotes.   
  
……………………………………………

If a coin has a Theo of $10 this should result in our quotes looking like this:

<https://docs.google.com/spreadsheets/d/1nD8nVzv0ATWUFlTkgzfIjbLyv9xV-OYH--4QwjQn1pE/edit?gid=0#gid=0>

| Fair Value | $10.00 |  |  |  |
| --- | --- | --- | --- | --- |
| bids |  |  | offers |  |
| quantity | price |  | quantity | price |
| 10.005003 | $9.995 |  | 9.995002 | $10.005 |
| 200.300451 | $9.985 |  | 199.700449 | $10.015 |

The trader (Kalon, Kate, Mark) should be able to enter an arbitrary set of bid quotes and/or offer quotes on a particular venue, with the 2 parameters (size and bps vs. Theo) and the quoting system should start quoting on that basis.

Edge configuration should always be positive if we’re collecting edge, so 5 bps entered on the bid side should mean we should bid at Theo \* 0.9995% and collect the 5 bps. If Edge is configured as a negative number (-5 bps) it means we should be bidding above Theo, so Theo \* 1.0005% for instance.   
  
……………………………………………  
  
We are using a Maker-Maker strategy, which becomes a Maker-Taker if needed to hedge.  
  
Step 1: we quote base level for $100 10 bps wide as a post-only order (guaranteeing maker fee) on the less liquid exchange (Hyperliquid), so -5 bps under bid and +5 bps over offer, relative to the Theo calculation.  
  
Step 2: while the quote is out there, we evaluate if we get hit on our bid scenario, where is the best place to sell out of Binance and ByBit. Also, calculate the scenario if our offer gets lifted.  
  
Step 2: we receive a fill on Hyperliquid, our bid gets hit. We send a Limit Order (could be filled as a maker fee or could hit a fresh bid as a taker) offer at the Theo calculated level to Binance or ByBit (whichever we decided on in Step 2).  
  
Step 3: if the hedge at Theo level Limit Order is not filled after 2 secs. Pull it and re-enter as an IOC to sell -5 bps under the current Theo value. Also, quotes should be skewed lower on all three venues by 2.5 bps (half the original fills amount).  
  
We do not want to simply use the skewed quotes to hedge our fill.  
  
The quotes are out there for the next potential trade. A hedge order is needed to hedge the initial trade.

We can easily analyse our data seeing the performance of our:

* post-only orders / initial quotes,
* our limit orders / hedge mid-point, and
* IOCs / aggressive hedge.

……………………………………………

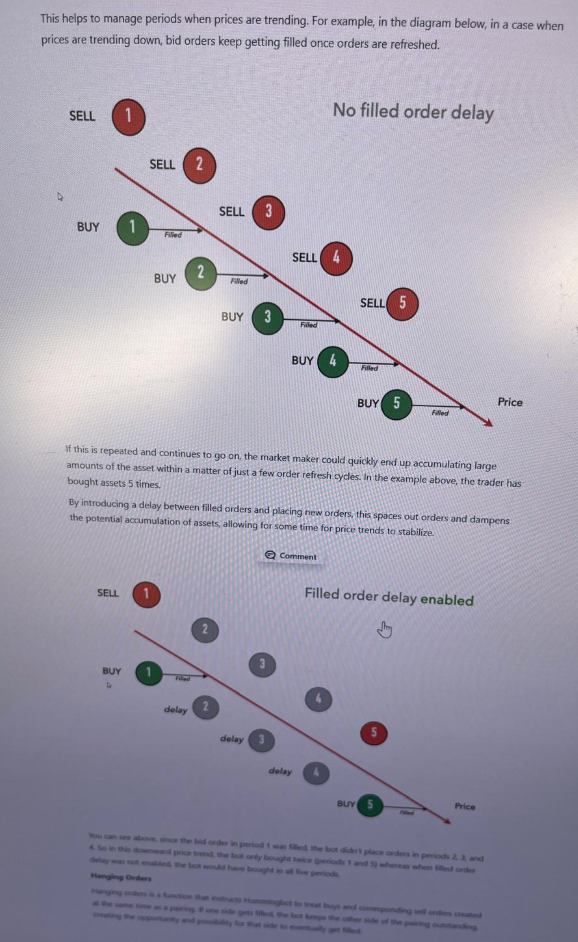
### When to cancel and re-enter:

### Two post-only orders are placed -5 bps under and -15 bps. If the Theo moves by 2 bps, modify the order, adjusting by 2 bps to re-centre it.

### Skew Module: (for MM quoting only)

When we are indifferent, we quote around like mentioned above. This could be if the funding arb is relatively small but still on our side.  
  
When we have a positional bias and are either happy to reduce or actively want to reduce, we will skew our quote on each of the three venues.  
  
**Scenario 1:** where the funding arb is small to flat, so happy to reduce.  
  
If we are short XXX\_p on Hyperliquid, we will skew our bid and offer higher. E.g. instead of quoting the $100 first order 5 bps under/over, we will quote it -2.5 bps under and +7.5 bps over.  
  
At the same time, say we are long XXX\_p on ByBit and flat position on Binance. On ByBit we will quote -7.5 bps at +2.5 bps.  
  
**Scenario 2:** now, we have the case where the funding is against and we are in the actively unwinding mode.  
  
We will only quote one-sided on each venue. We will bid XXX\_p on Hyperliquid at our Theo value and no offer (on the less liquid venue). Only once this is filled, we will send a Limit Order on ByBit to sell the Theo value with a Limit Order, if it doesn’t fill in 2 secs, pull it and send an IOC with -10 bps below Theo as a taker. Once Binance is included, we will assess whether the hedge should be on ByBit or Binance, since we may want to unwind Hyperliquid by legging into Binance, turning the trade into Binance vs. ByBit.

Skew should also be adjusted as we’ve been hit in accordance with these notes:



### Position Management:

Once an order has been filled, it is important to hedge quickly. We should adjust the skew of the quotes on the hedging venue rapidly, depending on the volatility of the coin.

We can measure real time volatility using some formula :(<https://chatgpt.com/share/67fcbcbd-9da4-8013-a161-7b78c19cfaae>)

Evgeny should decide on the method, given it’s about fast calculation, data, etc.  
Most volatility measures are close enough approximations for our purpose, but we want speed and no strain to the system.

If the real time volatility is too high, we should hedge immediately as a taker in the hedge venue’s orderbook. We should not place Market orders but should place Limit orders at the bid (if selling) price and leave them for 2 seconds and if not filled, change the price of the order to the new bid price - 10 bps.

If volatility is low we should simply skew our offers in the hedge venue to be closer to our Theo value, even Theo +1bp, for example.

The aggressiveness of this skew should be algorithmically determined and we need to decide how to do that.

……………………………………………

* Above we mentioned the base level quotes. Now we consider our quote widths when volatility picks up.
* We calculate the realised vol (RV) of the asset, over the past seven days. We work out our move buffer limit, Y.
* When the asset’s price moves by X% in one direction over the past 5 mins, which is our quote buffer, we widen our quotes accordingly.
* E.g. SOL has a realized vol of 75% avg. over the past seven days. This converts to an expected daily move of 3.9%.  
    
  Formula is: SQRT(365) = 19.1 (since 365 trading days)  
   75% vol / 19.1 = 3.9% (expected daily move given RV)  
    
  Therefore,

If X < 0.25Y no quote widening  
 If X > 0.25Y AND X <= 0.50Y, quote spread will be 1.5x.  
 If X > 0.50Y AND X <= 0.75Y, quote spread will be 2x.  
 If X > 0.75Y AND X <= 1Y, quote spread will be 3x.  
 If X > 1Y, quote spread will be 5x.  
  
 Need a manual override button to set our Zx level.

### Order Type Usage:

If Edge is configured to be a positive number, we should use the Post Only order type, so that we’re guaranteed to be a maker and price protected against being picked off.  
  
Post-only for opening order in the Maker-Maker and Maker-Taker.  
And Limit Order for the hedge leg. Therefore, by placing a mid-point for the hedge, there’s a chance we get maker, a chance we get taker, but most importantly, we’ll more likely have our hedge filled.

### Batch Place and Batch Modify:

As a market maker you will be placing dozens of orders every single second, so it is inefficient to do this via 12 or 24 limit orders per second for instance. It is better to do a single Batch Place and Modify Multiple Orders at once for example here: <https://hyperliquid.gitbook.io/hyperliquid-docs/for-developers/api/exchange-endpoint#modify-multiple-orders>

This will save on server costs but also be key to speed of trading.

Notes from call:

1.HLQ at midpoint (Post-only)2. You need to get filled on HyperLiquid3. Once filled on HQL, opposing order on ByBit : ICO at midpoint4. After 2 seconds, IOC at (midpoint -/+ 10 bps) for sell/buy

1.HLQ at midpoint (Post-only)

2. You need to get filled on HyperLiquid

3. Once filled on HQL, opposing order on ByBit : ICO at midpoint

4. After 2 seconds, IOC at (midpoint -/+ 10 bps) for sell/buy

………………

Kalon: I’ve made adjustments to the documents. On the call we were conflating the funding arb execution with the MM quoting. I have separated them out.

……………….