

Basket litepaper

One-click stock baskets on Robinhood Chain. Fixed share amounts, separate prices, clear stale states.

Chain 4663 · Rpc <https://rpc.mainnet.chain.robinhood.com> · WETH 0x0Bd7D308f8E1639FAb988df18A8011f41EAcAD73

1. Market

Tokenized real-world assets sit near \$46B, with about \$868M in related stable float. That is a small slice next to onchain meme volume, which still sets the pace for retail flows.

Equity exposure onchain starts from a thin base. Few stock tokens hold deep pools. Most size clears through quote desks rather than AMMs. Any basket product has to assume thin pools and build execution around RFQ from the start.

2. Problem

Four frictions repeat across stock tokens: a small RWA share near 4% of the relevant retail flow; off-hours trades near 51% of stock token volume landing when equity feeds are closed; thin AMM pools that slip on size; and dispersed single-stock picking that costs repeated transactions and quote checks.

Buyers either overpay for convenience or under-check staleness. Baskets address both by fixing amounts once and gating every price view on a staleness check.

3. Design

A basket stores fixed raw amounts per stock token at creation. Buying mints basket shares against those amounts. Redeeming burns shares and returns the stored raw amounts, rounded down.

Price is kept separate from share structure. Share amounts are fixed raw integers with 18 decimals. Display amounts divide raw by the ERC8056 uiMultiplier and are views only. Value multiplies raw by the Chainlink price (8 decimals, multiplier already included) and divides by 1e8. Redeem performs no oracle read.

Mint and burn are limited to approved participants. US persons are excluded. Feeds carry per-feed max ages, and any breach marks the row stale.

4. Math

Canonical formulas:

```
raw      = balanceOf(token, holder)    // uint, 18 dec
price    = chainlinkPrice(token)       // uint, 8 dec, includes multiplier
valueUsd = raw * price / 1e8
ui       = raw / uiMultiplier(token)   // display only
redeem   = floor(storedRaw * shares / supply) // no oracle read
```

Never compute display amount times price; that double-counts the scalar. Round redemptions down in favor of the basket. ETH is authoritative for launch math. Execution routes through RFQ desks (0x, 1inch, LiFi routing) as primary; AMM pools are consulted only when a quote shows they clear better.

5. Risks

Stale feeds: off-hours and halted feeds stop updating. The UI shows a stale badge, blanks the price, and excludes the row from fresh totals. Thin RFQ: desks can widen or pull size in stress, so large orders may split or rest. Corp actions: splits, dividends, and symbol changes alter the meaning of a fixed amount; each basket sets its own handling rules at creation. Jurisdiction: US persons are excluded and controls apply at mint, burn, and RFQ stages. Policy buyback: the protocol share funds a policy-only buyback that is not enforced onchain and can change or stop; it implies no floor price and no return.

6. Launch and roadmap

Launch runs through the active Pons factory on Robinhood Chain: factory 0xA5aAb3F0c6EeadF30Ef1D3Eb997108E976351feB at block 8991118; locker 0x736D76699C26D0d966744cAe304C000d471f7F35; supply 1,000,000,000; trading fee 1% split creator 70 / protocol 30; launch fee 0.0005 ETH; graduation 4.2 ETH same pool; block zero creator-only; two-block window with 5% holding cap and 5.5% buy cap.

Roadmap order: basket core with stale badges and RFQ execution, then the Pons launch, then sector baskets with reweight reviews and corp-action handling. Dates are set when each phase clears review.