



StonkBrokers Official Hooks

Pair token headliner (WETHNVDA and friends, uppercase on chain tickers), fee flows, co marketing and the countersigned Phase 1 agreement for the six official surfaces on Robinhood Chain.



In partnership with Hookr (hookr.fun) for Uniswap v4 hook launches

Date: 2026-10-13

Repository commit: 810c9aaf668a

Publisher: SB (BVI) Ltd, StonkBrokers

Contact: simple@clutch.market, Telegram simplefarmer69

Status: Battle, Strategy, Fractional 404 and Smart LP hooks are LIVE on Robinhood Chain 4663 (Sourcify exact match). Phase 1 is LOCKED IN: Hookr partner Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 receives partnerBps 5000 of the protocol cut; StonkBrokers signs setPartner and starts the ownership move to company Safe 0x6D6981459be6EB09C778d98527bCCc1a235bbb26. Mutual badges and co marketing of the launch are part of the go live. The Pair Token Hook (tickers are the uppercase currency order concat of both legs, e.g. WETHNVDA, AMZNPCX) is LIVE on Robinhood Chain 4663 since 2026-10-12 (factory, implementation, fee manager, launcher, hook; Sourcify exact match), tested 153 unit + 7 fork (160). Owner stays the StonkBrokers deployer until the user signals the Safe handover.

Plan status: Hookr replies 1 to 3 are locked. Phase 1 go live now. Phase 2 builds Hook Blocks versions together with the usage fee split recorded in Part E.

How to read this document

Part A is the directory brief. Part B is the functionality of each hook; the Pair Token Hook (WETHNVDA family) is the product headliner. Part C is the fee flow and the Hookr partner Safe. Part D is the Hookr UX integration guide (Universal Router on custom hooks). Part E locks the Hookr replies, fee governance, co marketing, dual day one WETHNVDA markets, Phase 1 / Phase 2 and the countersigned agreement with signature blocks. Per hook guides and ABIs live under apps/web/public/integration/hooks/.

Part A. Directory brief

A1. Official six



In partnership with Hookr

#	Name	Kind	Integration guide
1	StonkBroker Hook	Hookr RulesLaunch preset	StonkBrokerHook-Hookr-Integration.pdf
2	Smart LP Hook	Custom v4 hook + ladder	SmartLpHook-Hookr-Integration.pdf
3	Strategy Token Hook	Custom v4 hook + launcher	StrategyTokenHook-Hookr-Integration.pdf
4	Battle Token Hook	Custom v4 hook + vault (flagship)	BattleTokenHook-Hookr-Integration.pdf
5	Fractional 404 Hook	Custom v4 hook + NFT escrow	Fractional404Hook-Hookr-Integration.pdf
6	Pair Token Hook	Factory + ERC-20 pair vaults + preset launcher	PairToken-Hookr-Integration.pdf

Battle Token Hook is the priority product narrative for Hookr marketing: rival tokens, decaying snipe tax, permissionless epoch settle into a winner buy. Strategy and Fractional share the same StonkFeeHook base as Battle. Smart LP is the volatility farming surface with its own oracle anchored fee hook. The StonkBroker Hook is a preset on Hookr's own RulesLaunch root and uses Hookr's native rails. The Pair Token Hook (WETHNVDA family) mints one plain ERC-20 per pair (tickers are the uppercase currency order concat, e.g. WETHNVDA, USDGNVDA) that wraps a full range, auto compounding v4 position:

anyone can buy the token on any venue, deposit either side to mint it, or redeem it for the underlying pair, and a StonkFeeHook preset lets Hookr launch a new pair where the vault token is the only LP.

A2. Royalty and ops wallets

Role	Address	Notes
Royalty claim (StonkBroker preset)	0xb668382cf44038a3E8140E789060F6A809787CDa	Hookr royaltyTo for RulesLaunch. Owner of the claim flow on Hookr Treasury rails.
Protocol fee recipient (custom hooks)	0xb668382cf44038a3E8140E789060F6A809787CDa	protocolTo on BattleVault, StrategyLauncher and FractionalLauncher (and on PairTokenLauncher once deployed). Receives the StonkBrokers share of the protocol leg.
Hooks owner / deployer	0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C	Signs setPartner and transferOwnership. Pending owner is the company Safe until acceptOwnership.
Company Safe (2 of 3)	0x6D6981459be6EB09C778d98527bCCc1a235bbb26	Target owner of BattleVault, StrategyLauncher, FractionalLauncher and SmartLpHook after acceptOwnership.
Clock In booster (Pair Token carve)	0xB673276C46dE1428EacC15Ed6eceF480259D8566	Directed Clock In engine. Receives 50 percent of the Pair Token 10 percent compound carve through PairTokenFeeManager.flush.
STONKBROKER buyback wallet (Pair Token carve)	0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8	Receives the other 50 percent of the Pair Token compound carve. Same split as Smart LP.
Hookr partner Safe	0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9	Live partnerTo / partnerRecipient. partnerBps 5000 of the protocol cut.
Uniswap v4 PoolManager	0x8366a39CC670B4001A1121B8F6A443A643e40951	Canonical 4663 PoolManager. Hard pinned in scripts and guides.

Never send royalties or fees to a foreign chain copy of these addresses

Every address in this document is a Robinhood Chain 4663 fact. Code on other chains is not assumed, and funds sent to these addresses on any other EVM chain are not recoverable.

A3. Routing rule for the Hookr site

If a pool's hooks address is a Hookr owned root (StonkBroker RulesLaunch preset): use HookrRouter, HookrQuoter, and the Hookr SDK exactly as Hookr documents.

If a pool's hooks address is StonkFeeHook / SmartLpHook / any custom controller stack: do NOT call HookrRouter or HookrQuoter. Use Universal Router exactInput with empty hookData, and the Uniswap V4 Quoter for quotes. Directory cards should label these "vanillaSwap / Universal Router".

A4. Website checklist for Hookr

- List all six official names exactly as in A1 and render the StonkBroker badge beside each option.
- Render the Pair Token Hook card as the product headliner WETHNVDA (live on Robinhood Chain since 2026-10-12). Day one markets: WETHNVDA / HOOKR and WETHNVDA / \$STONKBROKER.
- Pin royalty claim wallet 0xb668382cf44038a3E8140E789060F6A809787CDa on the StonkBroker Hook card.
- Badge Battle Token Hook as the flagship product story beside the WETHNVDA headliner.
- On custom hook cards, show the Universal Router routing note (A3).
- Link each card to the matching *Hookr-Integration.pdf under the StonkBrokers integration folder.
- Point integrators at each guide's Contract functions section (Script commands for the preset).
- Link the Hashlock Certified hooks audit PDF from the security / docs strip.
- Do not mark custom hooks as Hookr owned roots.
- Co marketing: render mutual badges (StonkBroker on Hookr, Hookr on StonkBrokers) and Hookr frontend launch access for the listed surfaces.

- Partner Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 is already set on chain (Part C); no further address handoff needed.
- Custom hook cards can go live now (addresses in D1). Keep the StonkBroker preset launch CTA sealed until StonkBrokers unseals it (HARD HOLD).
- Phase 1 is agreed (Part E): list the hooks now as offered with the rev share. partnerTo on every controller, including the Pair Token launcher, already points at the Hookr Safe at 5000 bps.

Part B. Functionality of each hook

Each chapter follows the same shape: what the surface does, the lifecycle, the fee behaviour, the public functions an integrator calls, the events to index, and the guard rails. Function names are exact Solidity identifiers from hooks/src/. ABIs with full signatures ship beside this PDF.



B1. Battle Token Hook

StonkBroker official hook

Two rival tokens launch side by side, each behind a locked full range Uniswap v4 pool guarded by StonkFeeHook. Trading runs for one epoch (1 hour to 7 days, default 24 hours). When the epoch ends anyone can settle: the vault pulls a slice of the LOSER's liquidity, burns the loser tokens it took out, and spends the quote side plus the war chest on a winner buy that then deepens the winner's locked LP. Rounds repeat until a knockout (loser LP under floorBps or maxRounds reached).

Lifecycle

1. `createBattle(params)`: mints both BattleTokens, initializes both pools with the hook, seeds the full range LP from the creator's quote (at least `minSeedQuote` per side), starts the epoch clock and the decaying snipe tax.
2. Trading: exact input buys pay the base fee plus the decaying snipe tax; exact output while taxed reverts `ExactOutputWhileTaxed`; outside `addLiquidity` / `removeLiquidity` revert `LiquidityLocked`.
3. `settleable(battleId)` turns true when the epoch has elapsed. `settle(battleId)` is permissionless.
4. Settle: `leader(battleId)` picks the winner by pool price, `pullBps` (max 9000) of the loser LP is pulled, loser tokens burn, quote + war chest buy the winner through the winner pool, bought tokens and leftover quote are re added as locked winner LP, idle dust burns.
5. Knockout: once the loser LP sits under `floorBps` or `maxRounds` is reached the battle is final and trading on the loser pool stays open only for exits.

Fees

Every taxed swap leg splits: protocol leg `protocolBps` (1000 = 10%) of the swap fee, of which `partnerBps` (5000 = 50%) is carved for Hookr, the remainder goes to `protocolTo` (0xb668382cf44038a3E8140E789060F6A809787CDa); the controller leg funds the war chest and the creator cut. Base LP fee stays in the pool for the locked LP.

Public functions

Function	Who	Purpose
<code>createBattle(BattleParams)</code>	anyone	Launch two rivals, seed both pools, start the epoch
<code>settleable(uint256 id) view</code>	anyone	True when the epoch has elapsed and the battle is open
<code>settle(uint256 id)</code>	anyone	Pull loser LP, burn, buy winner, deepen winner LP
<code>battle(uint256 id) view</code>	anyone	Full battle struct (tokens, pools, round, war chest)
<code>currentSnipeBps(uint256 id) view</code>	anyone	Live decaying snipe tax in bps
<code>leader(uint256 id) view</code>	anyone	Current winner side by pool price
<code>battleOf(PoolId) / sideOf(PoolId) view</code>	anyone	Resolve a pool id back to its battle and side
<code>setProtocol(to, bps)</code>	owner	Protocol leg recipient / bps (max 2000)
<code>setPartner(to, bps)</code>	owner	Hookr partner recipient / share of the protocol leg
<code>setMinSeedQuote(uint256)</code>	owner	Floor for each side's seed
<code>setHook(address)</code>	owner	Pins the StonkFeeHook once
<code>BattleToken.burn(uint256)</code>	holder	Burn own tokens

Pool key

fee 10000 (1%), tickSpacing 200, hooks = Battle StonkFeeHook, currencies sorted by address. Pool id = `keccak256(abi.encode(PoolKey))`.

Events to index

BattleCreated, Settled, Knockout, ProtocolFeeTaken(poolId, quote, protocolCut), PartnerFeeTaken(poolId, quote, partnerCut, to), PartnerFlushed(quote, to, amount), PartnerSet(to, bps).



B2. Strategy Token Hook

StonkBroker official hook

A strategy token launches against a listed base asset into a locked full range pool. Taxed swap fees accumulate as pending quote; anyone can call execute to spend that quote on the configured strategy (buy the base, deepen LP or buy back the strategy token), tranced by impact and floored by slippage guards. Holders receive redeemable shares; a claim token tracks the distribution.

Lifecycle

1. createStrategy(params) returns (strategyId, token): mints the StrategyToken, initializes the pool with the hook, seeds the locked full range LP (at least minSeedQuote).
2. Trading accrues pending quote on the launcher per strategy.
3. execute(strategyId, minOut): permissionless; reverts NothingPending below minExecuteQuote, tranches by maxExecImpactBps, floors each tranche by maxExecSlipBps and the caller's minOut (SlippageExceeded(got, minOut) on a bad fill).
4. StrategyToken.notify streams the executed proceeds; holders use claimable / claim / claimTo.
5. withdrawLiquidity(strategyId) opens only after lpUnlockAt for the creator's own seed share.

Fees

Same StonkFeeHook split as Battle: protocol leg protocolBps 1000 with partnerBps 5000 carved for Hookr, remainder to 0xb668382cf44038a3e8140e789060f6a809787cda; the controller leg is the strategy's pending quote.

Public functions

Function	Who	Purpose
createStrategy(StrategyParams)	anyone	Launch strategy token + locked pool
execute(uint256 id, uint256 minOut)	anyone	Spend pending quote per strategy, tranced
withdrawLiquidity(uint256 id)	creator	After lpUnlockAt only
setExecGuards(slipBps, impactBps)	owner	Caps 2000 / 1000, defaults 300 / 100
setMinExecuteQuote(uint256)	owner	Dust floor for execute (1e15 live)
setProtocol / setPartner / setMinSeedQuote / setHook	owner	Same as Battle
StrategyToken.claimable(addr) view	anyone	Pending distribution for a holder
StrategyToken.claim() / claimTo(addr)	holder	Pull the distribution
StrategyToken.burn(uint256)	holder	Burn own tokens
StrategyToken.setExcluded(addr, bool)	launcher	Exclude pools / escrows from the stream

Events to index

StrategyCreated, Executed(strategyId, quoteIn, out), ProtocolFeeTaken, PartnerFeeTaken, PartnerFlushed, PartnerSet, StrategyToken Notified / Claimed.



B3. Fractional 404 Hook

StonkBroker official hook

An ERC-721 collection becomes a fungible market. Each deposited NFT mints a fixed 666,666 FractionToken units; burning that many redeems an NFT back. Fractions trade in a locked full range pool behind StonkFeeHook, and every vaulted NFT can get its own ERC-6551 account through the local registry so NFT level rewards keep landing while the NFT sits in the vault.

Lifecycle

1. createCollection(params) returns (collectionId, nft, token): wires the FractionalNFT wrapper + FractionToken, initializes the pool with the hook, seeds the locked full range LP.

2. `deposit(collectionId, tokenId)`: plain transferFrom pull (TBA safe, no onERC721Received needed), mints 666,666 fractions to the depositor.
3. `redeem(collectionId, tokenId)`: burns 666,666 fractions, releases the chosen vaulted NFT.
4. `deployAccounts(collectionId, n)`: permissionless paging that creates ERC-6551 accounts for vaulted NFTs (Fractional6551Account: owner / executeCall / executeTokenTransfer / isValidSignature).
5. `withdrawLiquidity(collectionId)` after the creator's LP unlock.

Fees

Same `StonkFeeHook` split: protocol leg `protocolBps 1000` with `partnerBps 5000` carved for Hookr, remainder to `0xb668382cF44038a3E8140E789060F6A809787CDa`; the controller leg funds the collection's buyback / LP deepening.

Public functions

Function	Who	Purpose
<code>createCollection(FractionalParams)</code>	anyone	Launch fractions + locked pool for a collection
<code>deposit(uint256 id, uint256 tokenId)</code>	NFT holder	Vault an NFT, mint 666,666 fractions
<code>redeem(uint256 id, uint256 tokenId)</code>	fraction holder	Burn 666,666 fractions, pull the NFT
<code>deployAccounts(uint256 id, uint256 n)</code>	anyone	Create ERC-6551 accounts for vaulted NFTs
<code>withdrawLiquidity(uint256 id)</code>	creator	After unlock only
<code>collection(id) / vaulted(id) / accountOf(id, tokenId) view</code>	anyone	Reads
<code>collectionOf(PoolId) / collectionOfNft(address) view</code>	anyone	Reverse lookups
<code>currentSnipeBps(uint256 id) view</code>	anyone	Live decaying snipe tax
<code>setProtocol / setPartner / setMinSeedQuote / setHook</code>	owner	Same as Battle
<code>ERC6551Registry.createAccount / account</code>	anyone	Deterministic account addresses

Events to index

`CollectionCreated`, `Deposited`, `Redeemed`, `AccountDeployed`, `ProtocolFeeTaken`, `PartnerFeeTaken`, `PartnerFlushed`, `PartnerSet`.



B4. Smart LP Hook

StonkBroker official hook

An oracle anchored dynamic fee hook for stock token pools. The swap fee is asymmetric: trades that push the pool AWAY from the Chainlink mark pay more, trades that bring it back pay the base fee. At the opening bell (feed resumes after a gap) the hook recaptures part of the stale price move as a surcharge. `SmartLpLadder` vaults deposit single sided stock inventory as a covered call style ladder and `SmartLpHookLens` exposes everything a UI needs in one call.

Lifecycle

1. Owner `registerPool(key, feed, stockIs0)` then `PoolManager.initialize(key, sqrtPrice)`: the pool carries fee `0x800000` (dynamic) and `tickSpacing 60`.
2. `quoteFee(key, zeroForOne)` returns (fee, oracleTick, poolTick, fresh, away) so a UI can preview the fee before the swap.
3. `beforeSwap` sets the dynamic fee; `afterSwap` skims the recapture / surcharge and books the LP, protocol and partner shares (Skimmed event).
4. `flush(key)`: permissionless; pays the partner recipient, then splits the remaining protocol share 50 / 50 between the StockBooster engine and the STONKBROKER buyback wallet.
5. `SmartLpLadder`: deposit stock tokens, the ladder mints single sided range positions above the mark and rolls them as fills land.

Fees

The base LP fee is 100% for LPs. Only the ADD ON income (recapture skim and surcharge) carries a 10% performance fee; `partnerBps 5000` of that protocol share is carved for Hookr, the rest splits 50 / 50 to StockBooster (`0xb673276C46dE1428EacC15Ed6eceF480259D8566`) and STONKBROKER buyback (`0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8`). Stock tokens are ERC-8056; all amounts are RAW units.

Public functions

Function	Who	Purpose
registerPool(key, feed, stocks0)	owner	Attach a Chainlink feed to a pool before initialize
quoteFee(key, zeroForOne) view	anyone	(fee, oracleTick, poolTick, fresh, away)
readMark(bytes32 poolId) view	anyone	Oracle tick + freshness for a pool
poolConfig(bytes32 poolId) view	anyone	Feed, stock side, params
flush(PoolKey key)	anyone	Pay partner, then booster / buyback split
setPartner(to, bps)	owner	Hookr partner recipient / share
requiredFlags() view	anyone	0x20c4 hook permissions
SmartLpHookLens.poolView(hook, key) view	anyone	Full UI struct incl. accruedPartner0 / 1
SmartLpLadder deposit / withdraw / roll	LP / anyone	Single sided ladder vault

Events to index

PoolRegistered, Skimmed(poolId, unspecified, lpShare, protocolShare, closed), Flushed, PartnerFlushed(poolId, partner0, partner1), PartnerSet(to, bps).



B5. StonkBroker Hook (RulesLaunch preset)

StonkBroker official hook

This surface is NOT a custom hook. It is a pinned configuration of Hookr's own RulesLaunch root, so it inherits Hookr's fee blocks (incl. Hookr's protocol share) and trades through HookrRouter / HookrQuoter like every other Hookr launch. StonkBrokers only pins the parameters and claims the royalty on Hookr Treasury rails.

Pinned parameters

Field	Value
royaltyTo	0xb668382cf44038a3e8140e789060f6a809787CDa
royaltyBps	1000
lpBps	300
burnBps	50
pot fields	0
seed	at least 95% of supply (5% hold cap)
ticks	full range +/-887220, spacing 60
currency0	ETH
Hookr protocol share	Hookr's own block, floor minProtocolShareBps 2000 to HookrTreasury

Hookr contracts used (HOOKR_1.contracts)

Contract	Address
Root	0x89c9FB50d1f03192230FFB1c279BB6E7d3Fe6aCc
Rules	0x4a71c270b1520F94064268102AebD744b0DE7942
Launcher	0x1cf2748A6670572425eAbf9Ab9e5BC4eb2eB8Ef4
Registry	0x463c82beA1a52F4Cc05ca0741a24F0AD24bA4602
Treasury	0x2242F7FC75606500503075f29ea8cDA684EfB4a9
FeeAdvisory	0x3B6b3f00684987B831fcF9EB5f0d5B806afF7306

Script: hooks/hookr/stonkbroker-hook.mjs with preflight / plan (read only, green against 4663) and launch / claim (HARD HOLD). hooklist entry kind rulesLaunchPreset.



B6. Pair Token Hook / WETHNVDA family (PRODUCT HEADLINER)

StonkBroker official hook



In partnership with Hookr

PRODUCT HEADLINER for the Hookr partnership. One plain ERC-20 per pair that wraps a full range, auto compounding LP position. The product name IS the on chain ERC-20 ticker: the uppercase currency order concat of both legs (WETHNVDA when WETH sorts as currency0, AMZNSPCX, WBTCWETH, NVDAUSDG). Day one markets now that the Robinhood deploy is live: WETHNVDA / HOOKR and WETHNVDA / \$STONKBROKER; AMZNSPCX next; pair token number six reserved. The vault wraps a full range, auto compounding Uniswap v4 position. The vault holds the position directly in the PoolManager through its own unlockCallback, so there is no position NFT and nothing a third party can pull. Anyone can buy the token on any venue, deposit either side (or ETH) to mint it, or redeem it for the underlying pair. Every pair uses the 1 percent pool (fee 10000, tick spacing 200) and each pair has exactly one token. PairTokenFactory creates vaults permissionlessly on hookless pools with a genesis seed in the same transaction; PairTokenLauncher is the Hookr preset that launches a new pair behind a StonkFeeHook where the pair token vault is the only allowed LP.

Lifecycle

1. Create: PairTokenFactory.create(tokenA, tokenB, amountA, amountB, sqrtPriceX96IfNew, minShares) initialises the hookless 1 percent pool when it does not exist, clones PairToken (EIP-1167), mints the genesis shares (1e3 dead shares burned) and returns the pair token address. One pair per pool.
2. Deposit: deposit(amount0, amount1, minShares, to) adds both sides in pool ratio; zapDeposit(tokenIn, amountIn, minShares, to) takes one side (or ETH) and swaps the balancing half inside the pool, bounded by the factory zap guards (ZapTooLarge / ZapMovesPrice).
3. Compound: every deposit and redeem collects pending fees first; compound() is permissionless and a keeper cranks it daily. Fees redeploy into the same position and the share price rises.
4. Redeem: redeem(shares, min0, min1, to) pays both sides pro rata; redeemZap(shares, tokenOut, minOut, to) swaps the other side out. A 10 bps retention stays with remaining holders.
5. Hookr preset: PairTokenLauncher.launch(token, quote, tokenAmount, quoteAmount, buyFeeBps, sellFeeBps, snipeBps, snipeDecaySecs, salt) seeds through the factory, hands the creator the share token, and the hook allowlists the vault as sole LP.

Fees

Mirrors Smart LP: 10 percent of fee revenue at compound (protocolFeeBps, owner capped at 2000) goes to PairTokenFeeManager, which pays the Hookr partner carve on preset pools (setPartner(poolId, to, bps), capped 5000) and splits the rest 50 / 50 between the Clock In booster 0xB673276C46dE1428EacC15Ed6ecef480259D8566 and the STONKBROKER buyback wallet 0xD2671d2Bdc84CF76E97ADb0fD7bb2B54F24CBDe8 on flush(token). Preset pools additionally carry the StonkFeeHook swap fee and snipe tax (buy / sell up to 1000 bps, snipe up to 9900 bps decaying over at most 6 hours), routed by PairTokenLauncher.onSwap to protocolTo and the partner.

Public functions

Function	Who	What
PairTokenFactory.create(...)	anyone	New pair vault + genesis seed, returns pairToken
PairTokenFactory.pairTokenFor(a, b) view	anyone	Resolve the one token for a pair
PairToken.deposit / zapDeposit	anyone	Mint shares, payable for native ETH
PairToken.redeem / redeemZap	holder	Burn shares for the pair or one side
PairToken.compound()	anyone	Collect fees, pay carve, redeploy
PairToken.basket() view	anyone	reserve0 / reserve1 behind the supply
PairToken.previewDeposit / previewRedeem	anyone	Quoter for the UI
PairTokenFeeManager.flush(token)	anyone	Pay booster / buyback split
PairTokenFeeManager.flushPartner(poolId, token)	anyone	Pay the Hookr carve

Function	Who	What
PairTokenLauncher.launch(...)	anyone	Hookr preset pair launch
PairTokenLauncher.currentSnipeBps(id) view	anyone	Live snipe tax

Events to index

PairCreated(pairToken, token0, token1, poolId, creator, hooked), Deposit, Redeem, Compounded(fees0, fees1, protocol0, protocol1, liquidityAdded), FeeNotified, Flushed, PartnerFlushed, PairLaunched, FeeForwarded.

Status: LIVE on Robinhood Chain 4663 since 2026-10-12 (hooks/test/PairToken.t.sol unit + hooks/test/PairToken.fork.t.sol fork, full ledger 160 / 160; deploy rail hooks/script/PairTokenDeploy.s.sol, 22 transactions, PHASE=verify clean, Sourcify exact match on all five contracts). Owner of the factory, fee manager and launcher stays the ops deployer until the user signals the handover. Seed anchors: native ETH, WETH, USDG, STONKBROKER and the ten Clock In 3.0 menu stocks, each with an owner set minSeed; any pair must include at least one anchor. Lead the Hookr card with WETHNVDA.

Part C. Fee flow and the Hookr partner leg

The Hookr fee exists in two different ways depending on the surface. The StonkBroker preset runs on Hookr's unmodified root, so Hookr's own protocol share block applies with no action needed from StonkBrokers. The four custom hooks are a new base authored by StonkBrokers, so a dedicated, owner settable Hookr partner leg was added to every controller and to the Smart LP hook, and the stacks were redeployed with it. Phase 1 sets partnerTo / partnerRecipient to the Hookr Gnosis Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 at partnerBps 5000. Accrued partner funds (if any) flush to that Safe; nobody including the owner can redirect them elsewhere. The Pair Token stack carries the same partner leg once deployed (PairTokenFeeManager / PairTokenLauncher setPartner).

C1. Split per surface

Surface	Leg	Who receives	Mechanism
StonkBroker preset	royalty 1000 bps	0xb668382cF44038a3E8140E789060F6A809787CDa	Hookr RulesLaunch royalty block
StonkBroker preset	protocol share >= 2000 bps	0x2242F7FC75606500503075f29ea8cDA684EfB4a9	Hookr's own protocol share block
Battle / Strategy / Fractional	protocol leg 1000 bps of swap fee	split by partnerBps 5000	StonkFeeHook.afterSwap carve
of which Hookr	5000 bps of the protocol leg	0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9	take to partnerTo
of which StonkBrokers	5000 bps of the protocol leg	0xb668382cF44038a3E8140E789060F6A809787CDa	take to protocolTo
Battle / Strategy / Fractional	controller leg	war chest / strategy / collection	controller logic + creator cut
Smart LP	base LP fee	LPs	stays in the pool
Smart LP	10% perf fee on add on income	split by partnerBps 5000	afterSwap skim
of which Hookr	5000 bps of the perf fee	0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9	accruedPartner0 / 1, paid on flush
of which StonkBrokers	remainder	StockBooster 50 / buyback 50	flush(key)
Pair Token (any pair)	base LP fee	pair token holders	compounds into the position
Pair Token (any pair)	10% carve at compound	split by partner bps (preset pools)	PairToken.compound notify
of which Hookr	partner bps (<= 5000)	setPartner(poolId, to, bps)	flushPartner(poolId, token)
of which StonkBrokers	remainder	Clock In booster 50 / buyback 50	flush(token)
Pair Token preset	swap fee + snipe tax	split by partnerBps 5000	StonkFeeHook afterSwap carve

C2. Hookr partner Safe (Phase 1 honor)

- Hookr partner Safe on Robinhood Chain 4663: 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9.
- The hooks owner 0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C signs setPartner(0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9, 5000) on BattleVault 0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304, StrategyLauncher 0x9f6821af226D85234DED5F0f024A8A2B4Fa6d023, FractionalLauncher 0xCD44629152c9a7CAc2cD68bb25A2293022F3183D and SmartLpHook 0x47137a7733E2B4DBD0e25416d5D5B29C8A1Da0c4. Four transactions, each emits PartnerSet. Agreed bps is 5000. Once the Pair Token stack is deployed the same owner (or the company Safe after accept) signs PairTokenLauncher.setPartner and PairTokenFeeManager.setPartner per preset pool.
- From the next swap on, StonkFeeHook takes the partner cut straight to the Safe (PartnerFeeTaken). The Smart LP hook accrues and pays on flush.
- Anyone may call StonkFeeHook.flushPartner(quote) to release any quote accrued before the address was set. Returns (to, amount) and emits PartnerFlushed.
- Anyone calls SmartLpHook.flush(key) per registered pool to release accruedPartner0 / 1 and the booster / buyback legs in

the same call.

6. Ownership: `0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C` signs transferOwnership(`0x6D6981459be6EB09C778d98527bCCc1a235bbb26`) on the four controllers and SmartLpHook. Company Safe owners then acceptOwnership (2 of 3).

Safety properties of the partner leg

partnerBps is bounded by 10000 (PartnerBpsTooHigh / BadParams). The partner cut is carved OUT of the protocol leg, never out of the LP fee or the controller leg, so creators and LPs are unaffected by the setting. Accrued partner funds have no owner withdraw path: they can only reach partnerTo / partnerRecipient. Pinned by hooks/test/Partner.t.sol and the SmartLpHook partner tests; whole suite 160 / 160 green (153 unit + 7 fork) incl. hooks/test/PairToken.t.sol and PairToken.fork.t.sol.

C3. Reading the partner state

```
# StonkFeeHook based stacks (Battle / Strategy / Fractional)
cast call 0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304 'partnerTo()(address)'
cast call 0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304 'partnerBps()(uint16)'
cast call 0x6ced3a0048F9f42836a90ee47FFe64645B0bAacc 'partnerAccrued(address)(uint256)' <quoteToken>
cast send 0x6ced3a0048F9f42836a90ee47FFe64645B0bAacc 'flushPartner(address)' <quoteToken> # permissionless o

# Smart LP hook
cast call 0x47137a7733E2B4DBD0e25416d5D5B29C8A1Da0c4 'partnerRecipient()(address)'
cast call 0x47137a7733E2B4DBD0e25416d5D5B29C8A1Da0c4 'partnerBps()(uint16)'
cast call 0x29Eaf335fed9270ba98BD5dd99519bb3af4C3F00 'poolView(address, (address, address, address, uint24, int24, address))'

# Pair Token stack (LIVE 2026-10-12)
cast call 0x1cEE10c87D261912A833E000f7e7295DC9E813E7 'partnerTo()(address)'
cast call 0x18AdF7be1d23022FcCf286D3d342d92Ecb3F9195 'partnerOf(bytes32)(address, uint16)' <poolId>
cast call 0x282c4815F30abc1D0A8eB810c7309140d35a5B38 'pairTokenFor(address, address)(address)' <tokenA> <tokenB>
cast send 0x18AdF7be1d23022FcCf286D3d342d92Ecb3F9195 'flushPartner(bytes32, address)' <poolId> <token> # perm
```

Part D. Integration guide for the Hookr UX

D1. Live addresses (Robinhood Chain 4663)

Stack	Controller	Hook
Battle Token Hook	BattleVault 0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304	0x6ced3a0048F9f42836a90ee47FFe64645B0bAacc
Strategy Token Hook	StrategyLauncher 0x9f6821af226D85234DED5F0f024A8A2B4Fa6d023	0x242FA7F53383a0e77Dd2A7778dC9Da794f042AcC
Fractional 404 Hook	FractionalLauncher 0xCD44629152c9a7CAc2cD68bb25A2293022F3183D	0x8A6c104d8bc04E058Ca08282E605D42a84D6aaCc
Smart LP Hook	SmartLpHookLens 0x29Eaf335fed9270ba98BD5dd99519bb3af4C3F00	0x47137a7733E2B4DBD0e25416d5D5B29C8A1Da0c4
Pair Token Hook	PairTokenFactory 0x282c4815F30abc1D0A8eB810c7309140d35a5B38	launcher hook 0x261a19357c66349BccEC55711F81612C0030aaCC

Support contract	Address
FractionalDeployer	0x3361426DEea55e88cb04cCA5cFeB46D99eB36dfF
ERC6551Registry (Fractional)	0x3Df336317B73e0ce417E03488abb2B79cEe2B9E4
Fractional6551Account impl	0xb9D00C31F479B892Ae7F8F500dbf3FF3E41acC26
Smart LP sINVDA ladder	0x3Ddf0412398D806AF1069B0EA29242b6BD22fd2B
Smart LP NVDA / USDG pool id	0x3b0a3db8c4d25fd428792aa6718796a3b87945e2b1d8b282500a753155a16e8b
NVDA (ERC-8056)	0xd0601CE157Db5bdC3162BbaC2a2C8aF5320D9EEC
USDG	0x5fc5360D0400a0Fd4f2af552ADD042D716F1d168
Uniswap v4 PoolManager	0x8366a39CC670B4001A1121B8F6A443A643e40951
Universal Router 2.1.1	0x8876789976dEcBfCbBbe364623C63652db8C0904
Uniswap V4 Quoter	0x8dc178efb8111bb0973dd9d722ebff267c98f94
Permit2	0x000000000022D473030F116dDEE9F6B43aC78BA3
WETH	0x0Bd7D308f8E1639FAb988df18A8011f41EAcAD73
CREATE2 deployer (hook mining)	0x4e59b44847b379578588920cA78FbF26c0B4956C
PairTokenFeeManager	0x18AdF7be1d23022FcCf286D3d342d92Ecb3F9195
PairTokenLauncher	0x1cEE10c87D261912A833E000f7e7295DC9E813E7
PairToken implementation	0x56C8c249d4c021e93635b120aC68A9a36eb01753

All nineteen deployed contracts are Sourcify exact match. Two earlier broadcasts of 2026-10-10 (zero launches each) are archived as `hooks/deployments/-hook.4663.superseded-2026-10-10.json` and `.superseded-2026-10-10b.json`; the Battle stack was redeployed 2026-10-13 for the HK-22 partial fill fix and its previous generation (zero launches) is archived as `battle-hook.4663.superseded-2026-10-13.json`; never list those addresses. The Pair Token stack (factory, implementation, fee manager, launcher, hook) went live 2026-10-12; state file `hooks/deployments/pair-token-hook.4663.json`, ledger `contracts/hooks/2026-10-12-pair-token-go-live.json`. Battle redeploy ledger: `contracts/hooks/2026-10-13-battle-partial-fill-redeploy.json`.

Chainlink feed	Address
NVDA / USD	0xC9d16E4f2569b9E3ea0468fD85844953713DC2a2
AAPL / USD	0xBb11A21267cFD63d4935d99a499133DD1744ACb
SPCX / USD	0xB265810950ba6c5C0Ff821c9963014a56fD8Bffb
TSLA / USD	0x4A1166a659A55625345e9515b32adECea5547C38

D2. hooklist entries

Suggested directory records. Flags are the hook permission bits encoded in each address: 0x2acc for StonkFeeHook (beforeInitialize, beforeAdd / RemoveLiquidity, before / afterSwap, before / afterSwapReturnsDelta) and 0x20c4 for SmartLpHook (beforeInitialize, beforeSwap, afterSwap, afterSwapReturnsDelta).

```
[
  { "name": "Battle Token Hook", "kind": "custom", "hook": "0x6ced3a0048f9f42836a90ee47FFe64645B0bAacc",
    "controller": "0x1BaDdAB75dd5223C2A619AA9c88BD211aaC5D304", "flags": "0x2acc", "routing": "universalRouter",
    "tags": ["battle", "snipe-tax", "locked-lp"], "badge": "stonkbroker" },
  { "name": "Strategy Token Hook", "kind": "custom", "hook": "0x242FA7F53383a0e77Dd2A7778dC9Da794f042AcC",
    "controller": "0x9f6821af226D85234DED5F0f024A8A2B4Fa6d023", "flags": "0x2acc", "routing": "universalRouter",
    "tags": ["strategy", "locked-lp"], "badge": "stonkbroker" },
  { "name": "Fractional 404 Hook", "kind": "custom", "hook": "0x8A6c104d8bc04E058Ca08282E605D42a84D6aaCc",
    "controller": "0xCD44629152c9a7CAc2cD68bb25A2293022F3183D", "flags": "0x2acc", "routing": "universalRouter",
    "tags": ["nft", "erc-6551", "locked-lp"], "badge": "stonkbroker" },
  { "name": "Smart LP Hook", "kind": "custom", "hook": "0x47137a7733E2B4DBD0e25416d5D5B29C8A1Da0c4",
    "lens": "0x29Eaf335fed9270ba98BD5dd99519bb3af4C3F00", "flags": "0x20c4", "dynamicFee": true, "routing": "u",
    "tags": ["stock-tokens", "oracle-fee", "covered-call"], "badge": "stonkbroker" },
  { "name": "StonkBroker Hook", "kind": "rulesLaunchPreset", "root": "0x89c9FB50d1f03192230FFB1c279BB6E7d3Fe6a",
    "royaltyTo": "0xb668382cF44038a3E8140E789060F6A809787CDa", "routing": "hookr", "badge": "stonkbroker" },
  { "name": "Pair Token Hook", "kind": "custom", "factory": "0x282c4815F30abc1D0A8eB810c7309140d35a5B38",
    "launcher": "0x1cEE10c87D261912A833E000f7e7295DC9E813E7", "flags": "0x2acc", "status": "live", "routing":
    "tags": ["full-range", "lp-token", "auto-compound"], "badge": "stonkbroker" }
]
```

D3. Quote and trade (custom hooks)

1. Resolve the pool: read battle(id) / collection(id) / the strategy struct for the PoolKey, or for Smart LP use the key the UI already holds (NVDA / USDG, fee 0x800000, tickSpacing 60).
2. Quote: Uniswap V4 Quoter quoteExactInputSingle with hookData empty bytes. For Smart LP call quoteFee(key, zeroForOne) first and show the dynamic fee.
3. Trade: Universal Router V4_SWAP with SWAP_EXACT_IN_SINGLE, SETTLE_ALL, TAKE_ALL. hookData empty bytes. Approvals through Permit2.
4. Never request exact OUTPUT on a Battle / Strategy / Fractional pool while the snipe tax is nonzero (ExactOutputWhileTaxed). Read currentSnipeBps(id) and switch to exact input.
5. Expect reverts from the hook, not the router: TradingPaused (Battle settle in flight), LiquidityLocked (outside LP), HookNotUsed (unused callbacks). Map them to plain copy.
6. Pair tokens trade on the hookless 1 percent pool through the Universal Router like any ERC-20. Mint and redeem go through the pair token contract itself: deposit / zapDeposit and redeem / redeemZap, quoted by previewDeposit / previewRedeem. Preset pools launched through PairTokenLauncher follow the Battle rules above (exact input only while the snipe tax is nonzero).

D4. Showing fees on the Hookr UX

- Custom hook cards: "Protocol fee 10% of swap fee, 50% of it to Hookr, the rest to StonkBrokers. LPs keep the base fee."
- Smart LP card: "Base LP fee 100% to LPs. 10% performance fee only on add on income, shared with Hookr."
- StonkBroker preset card: "Royalty 10% to StonkBrokers. Hookr protocol share per Hookr rules."
- Pair Token card: "Full range auto compounding LP token. 10% of fee revenue at compound, shared with Hookr on preset pools, rest 50 / 50 Clock In booster and STONKBROKER buyback."
- Index ProtocolFeeTaken / PartnerFeeTaken / PartnerFlushed / Skimmed to render lifetime fee tiles per pool. Pool ids resolve back to a battle / strategy / collection via the *Of(PoolId) views.

D5. Document package

Artifact	Path
This overview	docs/StonkBrokers-Official-Hooks-for-Hookr.pdf
Hashlock Certified hooks audit	docs/Hashlock-Certified-Audit-StonkBroker-Hooks.pdf
Per hook guides (MD + PDF)	contracts/hooks/integration/ and apps/web/public/integration/hooks/
Contract functions (per guide)	each *Hookr-Integration.md section Contract functions / Script commands
ABIs (18 files)	contracts/hooks/integration/*.abi.json (mirrored under public/integration/hooks/)
Deploy state	hooks/deployments/*-hook.4663.json
Pair Token deploy rail	hooks/script/PairTokenDeploy.s.sol (broadcast 2026-10-12, 22 txs, PHASE=verify clean)
StonkBroker preset script	hooks/hookr/stonkbroker-hook.mjs + README.md
Source	hooks/src/

D6. Deploy status and gates

Phase 1 honor LIVE: partner Safe set, ownership transfer to company Safe initiated

BattleVault, StrategyLauncher, FractionalLauncher and SmartLpHook are live on Robinhood Chain 4663 (Hashlock Certified hooks audit, HK-10 to HK-20 closed) with partnerBps 5000 and partnerTo / partnerRecipient = Hookr Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9. Ownership transferOwnership(0x6D6981459be6EB09C778d98527bCCc1a235bbb26) is initiated from 0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C; company Safe must acceptOwnership (2 of 3). The Pair Token stack is live (D1) with the Hookr Safe as partner at the same bps; only the StonkBroker preset ops launch stays on HARD HOLD until the user's explicit go. Re-run this builder after any hooks/src change. Plan status: Phase 1 locked in Part E (list now, rev share as built, co marketing, dual day one WETHNVDA markets now that the pair deploy is live). Phase 2 Hook Blocks + usage fee in Part E.

Part E. Hookr replies, co marketing and the locked Phase 1 agreement

The Hookr team reviewed this package across three replies. StonkBrokers agrees. This part reproduces the replies, locks the fee governance and Phase 1 honor steps StonkBrokers is executing on chain, records co marketing and the WETHNVDA headliner, and carries signature blocks for both parties. Commercial items that still need countersignature are marked PROPOSAL; Phase 1 technical honor (partner Safe, ownership move to company Safe, listing rails) is LOCKED IN.

E1. Hookr reply 1 (fees and Hook Blocks)

Hookr team, received by StonkBrokers

Hey man went through it, thank you for putting this together and baked in rev share. Quick q on the fees: how are the share and the payout address set up to change? Can the owner adjust them later, and is there any timelock or Safe on that side? We're on a wallet ourselves right now and moving to a Safe, so just curious how u're thinking about it. Also went through the hooks themselves, and most of it maps onto Hook Blocks we already run. Im happy to go live with what u offered as is now, ur hooks listed on hookr with the rev share, and if it goes well was thinking we could build out the Hook Blocks version together after: The StonkBroker preset already runs on our hook with LP Rewards, Auto Burn and the creator royalty. For Smart LP, Market Guard does the oracle surcharge and we have a market-hours fee. We'd just need to add an NVDA price feed and an oracle fee block. Strategy maps to Tax + Conversion and Zap Relay for buy-and-burn or buy-and-add LP. We'd add a guarded execute for the conversions. For Battle, Anti-Snipe and Tax + Conversion cover the fees. The epoch winner logic would be a new core module we'd build together. For Fractional, we can do the pool side. The NFT vault would stay on ur side as its own contract. And u'd own it. U can deploy it as ur own hook ur own contract and address, and u set the fees and which blocks run within each block's limits. On top of that they'd get arb recapture, which standalone hooks don't. U earn on every launch that comes through ur front end or partner code, and with the usage fee we're going live with on the Hook Blocks Market, u earn on every pool that runs ur blocks, whoever launches it. Let me know what you think.

E1b. Hookr reply 2 (partner Safe and listing)

Hookr team, follow up

Partner payout Safe on Robinhood Chain is 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9. Happy to list the six official StonkBroker surfaces now with the rev share as built. Keep the StonkBroker preset ops launch and any pair token seed on your HARD HOLD until you open them. Send the countersigned commercial pack when ready.

E1c. Hookr reply 3 (pair token headliner, co marketing, usage fee)

Quoted as received. The ethWnvda wording below is Hookr's shorthand; the final convention is the uppercase on chain ticker (WETHNVDA, AMZNSPCX) and the markets are named that way everywhere else in this document.

Hookr team, follow up

Love the ethWnvda headliner. Day one after your pair deploy go we want ethWnvda / HOOKR and ethWnvda / STONKBROKER markets. amznWspcx next; pair token number six stays your call. Co marketing: mutual badges (StonkBroker on Hookr, Hookr on StonkBrokers) and Hookr frontend launch access for the listed surfaces. Phase 2 usage fee on Hook Blocks Market we are going live with is roughly 45 / 25 / 20 / 10 (about 45 / 45 when HOOKR backs the pool). Lock Phase 1 now.

E2. Fee governance today (facts from the deployed bytecode)

Short answer: the share and the payout address are owner setttable with no on chain delay today. Phase 1 moves owner to the StonkBrokers company Safe so every fee setter needs 2 of 3. The carve for Hookr can only ever come out of the StonkBrokers protocol share, never out of the trader fee or the LP fee, and funds already accrued for Hookr have no owner withdrawal path.

Surface	What can change	Who / how	Bound
BattleVault, StrategyLauncher, FractionalLauncher (PoolController)	partnerTo and partnerBps via setPartner(to, bps). Takes effect on the next swap.	onlyOwner. Live partnerTo = Hookr Safe. Owner moves to company Safe after acceptOwnership.	bps <= 10000 of the protocol cut. Live value 5000 = half.
Same controllers	protocolTo and protocolBps via setProtocol(to, bps), plus setMinSeedQuote.	onlyOwner, same owner.	to != 0 (ZeroAddress), bps <= 2000 of gross swap fee (MAX_PROTOCOL_BPS). Live 1000.

Surface	What can change	Who / how	Bound
StonkFeeHook (per controller)	Nothing owner settable on the hook itself. It reads FeeTerms live from its controller.	flushPartner(quote) is permissionless and pays whatever partnerTo is at call time.	No owner path to accrued partner funds.
SmartLpHook	partnerRecipient and partnerBps via setPartner; booster and buyback via setRecipients.	onlyOwner. Partner share paid inside flush(key) before the 50 / 50 split.	Live partnerRecipient = Hookr Safe. partnerBps 5000.
PairTokenFeeManager (live)	Per pool partner via setPartner(poolId, to, bps); recipients via setRecipients.	onlyOwner (StonkBrokers deployer until the Safe handover).	bps <= 5000 (MAX_PARTNER_BPS).

Timelock: none in current bytecode. Safe: company treasury Gnosis Safe 2 of 3 at 0x6D6981459be6EB09C778d98527bCCc1a235bbb26 on Robinhood Chain 4663 is the Phase 1 owner target. protocolTo stays the royalty claim wallet 0xb668382cF44038a3E8140E789060F6A809787CDa.

LOCKED IN. Phase 1 fee and ownership honor

1) partnerTo / partnerRecipient set to Hookr Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 at partnerBps 5000 on BattleVault, StrategyLauncher, FractionalLauncher and SmartLpHook. 2) Ownership of those four controllers and SmartLpHook: transferOwnership(0x6D6981459be6EB09C778d98527bCCc1a235bbb26) from 0xD4aF10Bc56b08f8fc2990a3ADeeD4b757ae6A26C; company Safe owners acceptOwnership (2 of 3). 3) Fee change policy for countersignature: the Hookr share is never lowered without 30 days written notice to Hookr; the protocol share is never raised above its live value without the same notice; decreases of either may land immediately. Next bytecode generation carries this as an on chain delay. 4) Pair Token stack is LIVE (2026-10-12). Only the StonkBroker preset ops launch remains HARD HOLD until an explicit go.

E3. Phase 1. Go live as offered (LOCKED IN)

- Hookr lists the StonkBroker hooks on hookr.fun as offered in Parts A to D, with the rev share as built: Hookr receives 5000 bps of the StonkBrokers protocol cut on every pool under BattleVault, StrategyLauncher, FractionalLauncher and SmartLpHook, and the StonkBroker preset keeps the Hookr root share.
- Custom hook cards go live now with the addresses in D1; swaps route through the Universal Router (A3).
- PRODUCT HEADLINER: Pair Token card leads with WETHNVDA (live on Robinhood Chain since 2026-10-12). Day one markets: WETHNVDA / HOOKR and WETHNVDA / \$STONKBROKER. AMZNSPCX next; pair token number six reserved for a later StonkBrokers call.
- Co marketing: mutual badges (StonkBroker on Hookr, Hookr on StonkBrokers) and Hookr frontend launch access for the listed surfaces.
- The StonkBroker preset card goes live on Hookr's own rails. Launching it from StonkBrokers ops keys and seeding any hook pool from ops stay on HARD HOLD until the user's explicit go.
- Hookr partner Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 is set on chain (C2). Ownership transfer to company Safe 0x6D6981459be6EB09C778d98527bCCc1a235bbb26 is initiated; Safe acceptOwnership completes the move.
- Both sides sign the agreement block in E7. Until countersigned, commercial numbers beyond the live on chain config remain PROPOSAL.

E4. Phase 2. Hook Blocks versions built together

If Phase 1 goes well, the two teams rebuild each surface on Hookr's Hook Blocks so StonkBrokers launches sit on the shared core and gain arb recapture. Hookr's own mapping, which StonkBrokers accepts as the starting point:

Surface	Existing Hook Blocks	New work	Who builds
StonkBroker preset	Already runs on Hookr's hook with LP Rewards, Auto Burn and the creator royalty.	None. Keep the preset as the reference configuration.	Hookr (live today)
Smart LP Hook	Market Guard (oracle surcharge) and the market hours fee.	An NVDA price feed block and an oracle fee block. StonkBrokers supplies the Chainlink feed list and the surcharge curve from SmartLpHook.	Together

Surface	Existing Hook Blocks	New work	Who builds
Strategy Token Hook	Tax + Conversion and Zap Relay for buy and burn or buy and add LP.	A guarded execute for the conversions (minOut, impact tranches, keeper gate) mirroring StrategyLauncher.execute.	Together
Battle Token Hook	Anti Snipe and Tax + Conversion cover the fee side.	The epoch winner logic (net quote inflow per side, loser LP pull, winner buy and relock, knockout) as a new core module.	Together, new core module
Fractional 404 Hook	Hookr builds the pool side.	The NFT vault (ERC-404 + ERC-6551 at launch) stays a StonkBrokers contract and plugs into the Hookr pool side.	Hookr pool side, StonkBrokers vault
Pair Token Hook (WETHNVDA)	Not in the Hookr mapping yet (stack live on Robinhood Chain 2026-10-12).	Decide whether it joins Phase 2 or stays standalone.	Open

E5. Ownership, earnings and Phase 2 usage fee

- StonkBrokers owns the result: each surface deploys as its own hook, its own contract and address, under the company Safe after acceptOwnership.
- StonkBrokers sets the fees and chooses which blocks run, within each block's limits as published by Hookr.
- Pools on the shared core gain arb recapture, which the standalone hooks in Part D do not have.
- StonkBrokers earns on every launch that arrives through its front end or its partner code.
- Phase 2 usage fee on the Hook Blocks Market (Hookr reply 3, PROPOSAL until countersigned): roughly 45 / 25 / 20 / 10, about 45 / 45 when HOOKR backs the pool. Exact rate and settlement are Hookr's to publish and both parties countersign.

E6. Status of Phase 1 items

Item	Owner	Status
Hookr partner Safe for setPartner	Hookr	LOCKED IN 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9
setPartner + flushPartner on four controllers and SmartLpHook	StonkBrokers	EXECUTED Phase 1
Move hook ownership to company Safe (transfer + accept)	StonkBrokers	transferOwnership initiated; Safe acceptOwnership open
List six official surfaces with rev share as built	Hookr	Ready; listing rails in Parts A to D
Co marketing badges + Hookr frontend launch access	Both	LOCKED IN for Phase 1
WETHNVDA stack deployed 2026-10-12; day one WETHNVDA / HOOKR and WETHNVDA / STONKBROKER markets	StonkBrokers + Hookr	Live, markets open on seed
AMZNSPCX then pair token number six	StonkBrokers	After WETHNVDA
Fee notice policy (30 day before Hookr share lowered) and next bytecode delay	Both	PROPOSAL for countersignature
Hook Blocks Market usage fee 45 / 25 / 20 / 10 (~45 / 45 with HOOKR)	Hookr	PROPOSAL Phase 2
Phase 2 scoping call per surface (E4)	Both	After Phase 1 is live
Countersigned commercial terms (E7)	Both	Signature blocks below

E7. Agreement terms and signature blocks

The parties agree as follows for Phase 1 go live and integration. Items marked PROPOSAL still need countersignature before they bind commercially; the on chain Phase 1 honor steps do not wait on that.

1. Rev share as built: Hookr receives 5000 bps of the StonkBrokers protocol cut on BattleVault, StrategyLauncher, FractionalLauncher and SmartLpHook pools; StonkBroker preset keeps the Hookr root share on RulesLaunch.
2. Partner payout address is Hookr Safe 0x8Fdfaa25e2E4dDa297F301ea06458f22186D6bE9 on Robinhood Chain 4663.
3. Ownership of the four controllers and SmartLpHook moves to company Safe 0x6D6981459be6EB09C778d98527bCCc1a235bbb26 (transferOwnership then acceptOwnership).
4. The Hookr share is never lowered without 30 days written notice to Hookr (PROPOSAL until countersigned; next bytecode may encode the delay).

- 5. Co marketing: mutual badges and Hookr frontend launch access for the listed surfaces.
- 6. WETHNVDA is the Pair Token product headliner (uppercase on chain ticker convention); day one markets are WETHNVDA / HOOKR and WETHNVDA / \$STONKBROKER; AMZNSPCX next; pair token number six reserved.
- 7. StonkBroker preset ops launch remains HARD HOLD until StonkBrokers gives an explicit go; the Pair Token stack itself is live.
- 8. Phase 2 Hook Blocks rebuild and usage fee split are scoped after Phase 1 (E4, E5).

Signature blocks (print or DocuSign). Date: _____

Party	Name / title	Signature	Date
SB (BVI) Ltd (StonkBrokers)	Admir Zlatic, Founder	_____	_____
Hookr / Hookr.fun	Authorized signatory	_____	_____

Entity for StonkBrokers: SB (BVI) Ltd, Suite 5, Oleander Building, Port Purcell, Tortola, VG1110. Contact simple@clutch.market. Hookr fills its legal entity name above.