Vortex Wallet Register V2 Final

Protocol Whitepaper

Version 1.0

8 September 2026

Abstract

Vortex Wallet Register V2 Final is a smart contract on BNB Smart Chain that coordinates token-funded positions, USD-denominated reward accounting and VTR settlement. Participants contribute VTR and LVTR in a defined value ratio. The contract records positions, computes capped time-based accrual and allocates referral, leadership and pool rewards. Token pricing combines cumulative-price observations from liquidity pairs with a configurable fallback mechanism. V2 Final also provides state migration and administrative tools for maintaining referral structures and rank assignments. This paper describes the implemented rules, the movement of assets and the separation between accounting balances and the reserve used for payment.

1. Introduction

The protocol connects three activities: participation through token deposits, allocation of rewards under contract-defined rules, and settlement of eligible balances in VTR. Each activity is recorded through transactions and can be inspected using the contract's public state and events.

The contract uses USD as an accounting unit. An amount recorded in USD is a calculation basis, not a holding of US dollars or a claim on a USDT escrow. Actual contributions use VTR and LVTR; ordinary withdrawals and pool payments use VTR.

This paper describes the V2 Final implementation at the address in section 8. Configurable tables are identified as deployment defaults. Numerical examples use illustrative prices and are not market quotes or projections.

2. System architecture

The register maintains participant accounts, referral relationships, positions, reward balances and pool entitlements. VTR and LVTR are external token contracts. WBNB and USDT provide the quote assets for the pricing system.

Deposit tokens go to an immutable collector address. A separate VTR balance held by the register supplies withdrawals and pool payments. The brotherhood address receives the designated withdrawal fee. Owner and account-manager roles maintain the settings and account operations described in section 7.

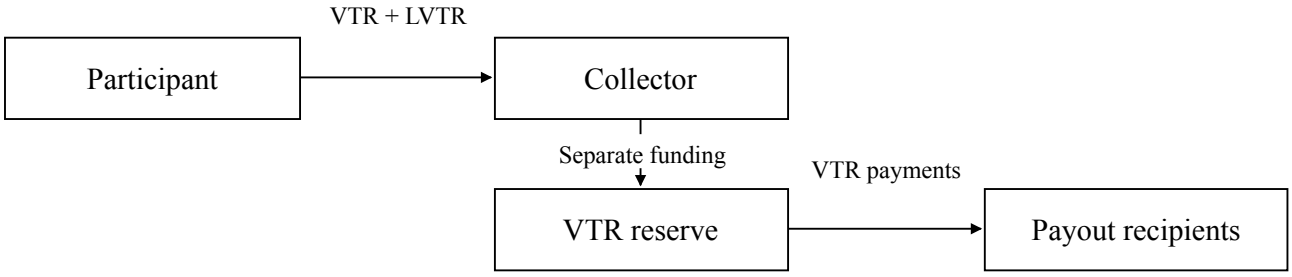

Figure 1. Deposit collection and reserve funding are separate operations. Pool allocations remain reserved inside the contract until claimed.

3. Deposits and positions

3.1 Registration and contribution

After migration is sealed, an address may register once, either with an existing registered sponsor or without a sponsor. Self-referral is rejected. A registered participant may create multiple positions; each deposit is accounted for independently.

The minimum deposit value is $100. For a requested USD value D, 70% is contributed in VTR and 30% in LVTR. If P_V and P_L are the respective USD prices per token, the whole-token quantities are:

$$VTR_{in} = 0.70D / P_V \quad LVTR_{in} = 0.30D / P_L$$

The participant approves both tokens and supplies maximum acceptable input amounts. The deposit reverts if either required amount exceeds its limit. Successful transfers send both assets to the collector and create a position containing its USD value, start time, last claim time, cumulative ROI paid and active status.

Example 1 — A $1,000 position. At illustrative prices of $0.025 per VTR and $1.00 per LVTR, the contribution is 28,000 VTR plus 300 LVTR. The recorded position value is $1,000. The contract does not purchase these tokens for the participant.

3.2 Time-based accrual

Each position has a daily rate determined by its own USD-value tier. Additional positions do not move an existing position into a higher tier. The rate is calculated from a deterministic hash of the UTC day index and tier, so all positions in the same tier share that day's rate.

Table 1. Position tiers and inclusive daily rate ranges.

Position value	Tier	Daily rate
$100 ≤ D < $1,000	0	0.60–0.80%
$1,000 ≤ D < $5,000	1	0.60–1.00%
D ≥ $5,000	2	0.60–1.20%

In basis points, the rate is 60 + H(day, tier) mod (span + 1), where the span is 20, 40 or 60. One basis point is 0.01%. The hash makes the schedule reproducible; it is not a source of unpredictable randomness.

Accrual is proportional to elapsed seconds within each UTC day. A claim crossing a day boundary sums the corresponding daily portions. Accrual uses the original position value and does not automatically compound.

$$\text{Daily portion} = D \times rate_{bp} \times \text{elapsed seconds} / (10{,}000 \times 86{,}400)$$

3.3 Cap and settlement

Cumulative ROI for a position is capped at 2.5 times its original USD value. Claims credit the participant's USD balance. When settlement reaches the cap, the position becomes inactive. The original contribution is not a separate principal balance that can be withdrawn.

Participants may settle all positions or a selected range. Ranged claims divide the processing cost of large accounts across transactions. The minimum ROI credit per successful claim is $0.0001.

Example 2 — One day of accrual. If a $1,000 position has a 0.80% rate for a full UTC day, it accrues $8 for that day, provided at least $8 remains below its $2,500 ROI cap. Claiming records this amount as USD balance; receiving VTR requires a withdrawal.

4. Reward mechanism

4.1 Direct referral rewards

Deposits credit the direct sponsor with 5% of their USD value. Higher rates can unlock during a 72-hour accelerator window, normally starting with the sponsor's first deposit.

Table 2. Accelerator thresholds. Either condition in a row is sufficient.

Referral rate	Active direct referrals	Direct deposit volume in window
5%	Base rate	Base rate
7%	3	$5,000
8%	5	$10,000
10%	10	$25,000

The highest qualifying rate applies and does not subsequently decrease. Volume counts deposits during the window; active directs are counted by current active positions. The manager may reopen the window (section 7).

4.2 Matching rewards

ROI claims generate matching credits for up to ten ancestors. Each active direct unlocks two levels, capped at ten with five active directs. Eligibility is evaluated when the claim is processed.

Table 3. Matching percentages at deployment; configurable by the owner.

Upline level	1	2	3	4–5	6–10
Share of claimed ROI	5%	3%	2%	1.5% each	1% each

Matching credits do not reduce the claimant's ROI. Skipped ancestors do not compress levels. Active status updates on settlement; elapsed time alone does not remove a direct from the stored count.

4.3 Team volume and rank

Deposits update volumes for up to 50 ancestors. The big leg is the largest direct branch; small-leg volume is total direct-branch volume minus that branch. These measure cumulative contributions, not withdrawable balances.

Table 4. Rank configuration at deployment, not a live configuration snapshot.

Rank	Small-leg volume	Personal deposits	Leadership rate	Pool weight
1	$5,000	$100	5%	0
2	$30,000	$100	10%	0
3	$100,000	$500	15%	1
4	$500,000	$1,000	25%	2
5	$1,000,000	$3,000	35%	3
6	$3,000,000	$5,000	40%	5
7	$5,000,000	$10,000	45%	8
8	$10,000,000	$20,000	60%	10
9	$20,000,000	$20,000	70%	15

Both volume thresholds must be satisfied for an earned rank. Earned rank advances when refreshed and does not decrease. The effective rank used for leadership and pool weight is the higher of earned rank and any manager-assigned manual rank.

4.4 Leadership allocation

Leadership rewards are calculated on claimed ROI over a maximum of 50 ancestors. At each ancestor, the branch through which the claim originates is identified. The ancestor is excluded from leadership allocation for that claim if the originating branch is its recorded big leg.

For an eligible ancestor, the contract reads the leadership percentage associated with its effective rank. If that percentage exceeds the highest percentage already allocated earlier in the traversal, the ancestor receives the difference applied to the original ROI claim.

$$\textit{Differential reward} = \textit{claimed ROI} \times (\textit{eligible rate} - \textit{highest rate already allocated})$$

For example, a 15% eligible rate following a 5% allocation receives a 10% differential. This is a difference between configured percentages, not an additional 15% on top of the earlier allocation.

The implementation also provides two override cases. An ancestor at rank 3 or higher whose configured rate equals the highest rate already reached can receive 10% of the previous nonzero leadership bonus. An ancestor at rank 5 or higher whose rate is lower than that maximum can also receive 10% of the previous nonzero bonus. The comparison uses configured percentages, which need not follow rank order after configuration changes.

These overrides apply to the previous bonus, not directly to the original ROI amount. After a nonzero bonus is credited, it becomes the reference for the next applicable override. Leadership allocations are credited to USD balances.

4.5 Weekly VTR pool

Ordinary withdrawals allocate 2% of gross VTR to the pool. This amount stays in the register and increases both the undistributed pool amount and the protected pool reserve.

The pool operates on fixed seven-day cycles. On the first cycle-processing transaction after a boundary, accumulated tokens are allocated using the total pool weight recorded at that time. If there is no eligible weight, the amount remains accrued for a later cycle. Cycle processing is transaction-driven; the contract does not execute itself at a scheduled time.

$$\textit{Allocation per unit of weight} = \textit{distributable VTR} / \textit{total eligible weight}$$

The accounting uses a cumulative amount per unit of weight, together with each participant's recorded debt and retained credit. Before an individual's weight changes, already allocated entitlement is preserved. This avoids iterating over all participants to distribute each cycle.

A pool claim pays VTR directly and reduces the protected reserve. It does not first become a USD reward balance and does not apply the ordinary withdrawal fee. Integer division can leave rounding remainders for subsequent accounting.

4.6 Relationship between reward components

The 250% position cap applies to that position's ROI. Direct, matching and leadership rewards have their own allocation rules and are not deducted from the originating position's remaining ROI cap. Pool entitlement is maintained separately in token units.

A participant's USD balance may therefore contain credits from multiple sources. The balance is reduced by ordinary withdrawals; it is not automatically reinvested into new positions. None of these accounting operations creates external token liquidity by itself.

5. Token pricing

5.1 Quote paths

The price of VTR or LVTR is obtained through two quote legs: token to WBNB, then WBNB to USDT. The USDT-denominated result is used as the contract's USD accounting price. The protocol therefore uses USDT as its dollar reference; it does not read a separate fiat exchange-rate feed.

$$P_{token,\ USD} = TWAP(token / WBNB) \times TWAP(WBNB / USDT)$$

Each configured liquidity pair must contain the intended token and the required quote token. Token-pair feeds use WBNB as the other asset; the WBNB feed uses USDT. The implementation reads cumulative prices and reserves from Pancake-style pairs.

5.2 Observation window

A feed retains up to eight observations in a circular buffer. At most one observation is recorded per UTC day. Public maintenance calls and selected participant operations sample the feeds, so observation updates depend on transactions reaching the contract.

To calculate a time-weighted average price, the contract takes the difference between the current cumulative price and the oldest retained observation, then divides by elapsed time. The observation must be at least six days old for the feed to return a usable quote.

The six-day requirement is a minimum elapsed window, not a requirement for exactly six samples. If observations are sparse, the effective averaging period can be longer. Current cumulative values are extended from pair reserves and the last reserve timestamp where applicable.

5.3 Fallback pricing

A nonzero mature quote from both legs takes precedence. If the TWAP path cannot provide a nonzero result, the contract may use the configured fallback price for the requested token, provided fallback remains enabled and that configured price is nonzero. Without an available price, the price-dependent operation reverts.

The fallback is a configuration mechanism that supports the observation warm-up period. The owner or account manager may set its values while it is enabled. A fallback flag being enabled does not mean every current quote uses fallback.

The owner can permanently disable fallback once the VTR, LVTR and WBNB feeds all return nonzero mature quotes. This also prevents subsequent feed reconfiguration. The control is one-way: the implementation has no function to re-enable fallback.

5.4 Pricing and execution

A deposit converts requested USD value into VTR and LVTR quantities. A withdrawal converts an existing USD balance into gross VTR. Participant-specified maximum inputs and minimum output amounts constrain those executions.

The register does not execute a swap on these pairs as part of a deposit or withdrawal. The pairs supply pricing observations; actual token movements follow the collection and reserve paths described in sections 2 and 6.

Prices depend on the configured pairs, their liquidity, observation history and, when applicable, fallback configuration. A six-day average responds differently from an immediate spot quote. The formulas in this paper use whole-token quantities for readability; the implementation uses integer arithmetic and 18-decimal scaling.

6. Withdrawals and reserves

6.1 Converting the reward balance

A participant can request an ordinary withdrawal of at least $10, up to their recorded USD balance. The contract determines the gross VTR amount from the current contract price and divides it between the participant, the brotherhood and the pool.

$$Gross\ VTR = requested\ USD / P_V$$

Table 5. Allocation of an ordinary withdrawal.

Recipient or allocation	Share of gross VTR	Destination
Participant	Approximately 95%	Participant wallet
Brotherhood	3%	Configured brotherhood wallet
Weekly pool	2%	Reserved inside the register

The implementation rounds each fee down to token base units and assigns the remainder to the participant. Its minimum-output check applies to the net VTR amount. Network transaction fees, paid in BNB, are separate from these allocations.

Example 3 — Withdrawing $100. At an illustrative VTR price of $0.025, the gross withdrawal is 4,000 VTR. The participant receives 3,800 VTR, the brotherhood receives 120 VTR, and 80 VTR remains reserved for the pool. The participant's USD balance decreases by $100.

6.2 Protected pool reserve

The payable reserve is the VTR balance held by the register minus the amount reserved for pool entitlements, floored at zero. Before an ordinary withdrawal, this payable reserve must cover the full gross VTR amount, including the new pool allocation.

$$Payable\ reserve = max(VTR\ held\ by\ register - pool\ reserve,\ 0)$$

This separation prevents ordinary withdrawals from spending tokens already designated for pool claims. During withdrawal, the USD balance and reserve accounting are updated before outbound token transfers. The entry point uses reentrancy protection.

6.3 Funding and settlement conditions

Reserve funding is a separate transfer of VTR into the register. The public funding function allows a token holder to approve and contribute VTR. Deposit collection does not automatically move the collector's assets back into the payout reserve.

Accrued ROI, credited USD balances and available VTR are distinct quantities. A claim can record an entitlement while its eventual token payment remains dependent on an available price and sufficient payable reserve. If the reserve check is not satisfied, an ordinary withdrawal reverts without consuming the participant's USD balance.

The register defines an allocation and settlement mechanism. It does not itself establish an external revenue source or guarantee future reserve coverage. The daily percentages specify contract accounting rules; successful settlement also depends on token availability and the execution conditions described above.

7. Governance and migration

7.1 Configuration roles

The implementation separates owner configuration from account-management operations. Token addresses and the collector are immutable. Other settings follow the access rules below.

Table 6. Administrative capabilities and constraints.

Operation	Role	Constraint
Update manager	Owner or manager	The role may be transferred or set to zero.
Configure rank thresholds and leadership rates	Owner or manager	Rank 1–9; rate ≤100%; pool weights freeze after sealing.
Configure fallback values	Owner or manager	Fallback must remain enabled.
Configure feeds; disable fallback	Owner	Feed changes require enabled fallback; disabling requires mature feeds.
Update brotherhood; matching rates	Owner	Nonzero brotherhood; each matching level ≤10%.
Move referrals; assign rank; reset accelerator	Manager	Registered accounts; migration must be sealed.

Ownership transfer uses a two-step acceptance process. Manager authority is a separate role and remains available if ownership is renounced. At the documented block 120,676,009, the owner was `0x6F5A0083CaE993B21Ad77034dCb00ea484A3c6e5` and the manager was `0xEfbd812a7c654cb035871a2E829DdfFb5B4CE28B`. This is a dated state record, not a live status display.

7.2 Account maintenance

A manager-driven referral change moves an account's subtree and adjusts the old and new ancestors' branch volumes and active-direct counts. Self-referral, cycles through the moved account and overly deep new sponsor paths are rejected. At depths beyond the 50-ancestor propagation boundary, aggregate subtree moves can differ from replaying individual deposits.

Big-leg rescans within a move share a 1,000-branch budget. A branch that cannot be fully rescanned emits a maintenance event; a public refresh can recompute the largest leg. Earned ranks remain persistent when a branch leaves.

A manual rank can raise effective rank above earned rank and can later be changed or revoked. Once earned rank catches up, the manual grant is cleared. Accelerator resets reopen the 72-hour window, retain accumulated window volume and do not reduce an unlocked rate. A reset before the first deposit starts the clock early.

7.3 Migration boundary

Before sealing, the owner may import referral structure and per-account positions, balances, contribution volumes and accelerator state. The importer deliberately omits ranks and pool entitlements under the source-snapshot assumption that these are zero; this is not a general migration mechanism for arbitrary nonzero pool state.

Sealing checks the expected structure import count and permanently closes import operations. Ordinary registration, deposits and reward claims then become available. Complete reconciliation of imported inputs remains part of the migration process. Migration was sealed at the documented block.

8. Technical parameters and references

Table 7. Implementation identity and fixed parameters.

Network	BNB Smart Chain mainnet; chain ID 56
Contract	VortexWalletRegisterV2Final
Compiler	`v0.8.35+commit.47b9dedd`
Build settings	Optimizer enabled; 1 run; via IR; Paris EVM target
Deployment	26 August 2026, 19:10:05 UTC; block 118,245,276
Minimum deposit / withdrawal	$100 / $10 in contract accounting units
Contribution value split	70% VTR / 30% LVTR
Position ROI cap	250% of the position's original USD value
Accelerator / pool cycle	72 hours / 7 days
Matching / leadership traversal	10 levels / 50 ancestors
Oracle window / observation capacity	Minimum 6 days / 8 observations per feed
Ordinary withdrawal allocations	3% brotherhood; 2% pool; remainder to participant

Register
`0x715EeB0DA4586a7c803301d8b304AfA234d9f803`

VTR / LVTR
`VTR: 0x620833798C1596C5C8201a0b2C2d36E992A17Efb`
`LVTR: 0x2dCb495f5BDF5BFc06Aa866710d1c7d118b9631a`

Quote assets
`WBNB: 0xbb4CdB9CBd36B01bD1cBaEBF2De08d9173bc095c`
`USDT: 0x55d398326f99059fF775485246999027B3197955`

The formulas use readable whole-token and USD units. Contract execution uses integer arithmetic with rounding. Tables 3 and 4 describe constructor defaults; current configurable values can be inspected through the contract's public getters. Token issuance and token-specific mechanics are defined by the respective token contracts.

References

1. Vortex Wallet Register V2 Final. Contract address, source and public interface.
   https://bscscan.com/address/0x715EeB0DA4586a7c803301d8b304AfA234d9f803#code
2. Deployment transaction. BNB Smart Chain, block 118,245,276.
   https://bscscan.com/tx/0xfae6dfdee7f3671f3f509a4be2501f6efe4cf0c8988af76f13cd21f51cfd8217
3. Documented state block. 8 September 2026, 11:11:06 UTC.
   https://bscscan.com/block/120676009

This whitepaper describes the deployed contract's mechanisms and configuration model. The deployed implementation and its state determine execution. The document does not replace the contract or constitute an independent security certification.