ATLAS SYSTEM

SMART CYCLE 2

Smart Cycle 2: Smart Contract and Infrastructure Registry

> Independent Smart Cycle 2 audits are in progress. Reports will be published when completed.
> The contract sends 2% to the designated recipient. Its code does not itself make compensation payments; the stated purpose of the recipient was supplied by the project team.

Contents

- 1. Verification Summary
- 2. Atlas Contract Registry
- 3. Shared Liquidity Infrastructure
- 4. Ownership and Administrative Control
- 5. LP-NFT Ownership and Operator Approvals
- 6. Interactions and Flow of Funds
- 7. Independent Verification Checklist
- 8. Fixed and Dynamic Values
- 9. Full Address and Link Index
- 10. Registry Update Rules
- 11. Important Explanation

Technical registry · Atlas System document library · 19 October 2026

Contract addresses, governance structure, liquidity infrastructure and independent on-chain verification. Contracts · Liquidity · Verification.

The registry was updated on 19 October 2026 with the SC2 main contract, deployment wallet, limit-management address, fee and fund recipients, and LP position No. 7682306. The project team supplied the stated purposes of the service addresses.

Smart Contract and Infrastructure Registry

Parameter

Value
Network BNB Smart Chain · Chain ID 56
Settlement asset USDT; verify the precise address using `token`
Explorer BscScan
Edition SC2 / 19 October 2026
Reference block and time 128510916 · 2026-10-19T07:03:19+00:00

What was retained from the original document

The full set of sections: registry, liquidity, management, LP-NFT ownership, flow of funds, verification methodology, fixed and dynamic values, address index and change log.

The description was updated using the final SmartCycle2 sources and ABI supplied on 19 October. The fund address was checked using `insuranceWallet()`, and the system sponsor using `SYSTEM_SPONSOR()`. The executable bytecode was reproduced using compiler solc 0.8.20 with the supplied settings and matched byte for byte after substitution of the verified `manager` address. This checks source correspondence; it is not an independent audit.

1. Verification Summary

Smart Cycle 2 combines cycle creation, payout calculations and partner calculations in the SmartCycle2 contract. Lockup Flow and Daily Flow describe cycle modes; separate addresses of the former Flow contracts are not used as SC2 addresses.

Configuration to verify

1. The contract is deployed on the correct network and its source code has been compared with the bytecode.
2. `token`, `treasury`, `manager` and `insuranceWallet` match the registry. `SYSTEM_SPONSOR` identifies the system sponsor for new registration with a zero `sponsor`.
3. `lpManager`, `lpFactory`, `lpPool` and `lpTokenId` point to one consistent LP position.
4. The LP-NFT belongs to SmartCycle2. No third-party address is approved for the individual NFT.
5. The position's token composition, fee and tick range match SC2's immutable parameters.
6. The manager's permissions match the code; completion of migration and the status of new cycle admission have been checked separately.

Main independent checks

1. Open the full SmartCycle2 address in BscScan and check the code and constructor parameters.
2. In Read Contract, read the basic addresses and LP parameters.
3. In Position Manager, check `ownerOf(lpTokenId)`, `getApproved(lpTokenId)` and operator approvals.
4. Reconcile transactions with cycle, payout and liquidity movement events.
5. Record the same block number and UTC time for all results.

These steps describe the methodology for a full review. Selected confirmed on-chain values are given in Section 9; publishing addresses does not replace an audit.

2. Atlas Contract Registry

2.1 Participant contract

Component

Purpose

Address

SmartCycle2

Cycle creation, payout claims, recording the structure and partner calculations.

`0x7E7BeDfcea906A1feF682ecb4e316F6bD87587E3`

Lockup Flow / **Daily Flow**

Modes within SmartCycle2, checked against the tariff and the individual cycle record.

Same SC2 address

2.2 Service roles and external contracts

Field / **component**

Purpose

`token`

Settlement token address. Read `decimals()` from the token contract itself.

`treasury`

Platform fee recipient.

`manager`

Address with limited permissions to announce limit policies and period opening times.

`insuranceWallet`

Recipient of the fixed 2% allocation to the SC1 fund. Value verified by reading the contract.

`lpManager`

External LP-NFT management contract.

`lpFactory` / `lpPool`

Factory and specific pool associated with the LP position.

SC2OwnLP, SC2CycleLedger and SC2PartnerSettlement are SmartCycle2 source-code modules. Their names do not imply separate published addresses. Section 9 contains the full address index and fields to verify. The old Transport and Distribute components are not included in the SC2 scheme without confirmation that they exist in the final deployment.

3. Shared Liquidity Infrastructure

Item

What to record and check

SC2 LP position

`lpTokenId` and `ownerOf(lpTokenId)`.

Position Manager

`lpManager` and its relationship to `lpFactory`.

Pool

`lpPool` and the result of `factory.getPool(token0, token1, fee)`.

Position tokens

`lpToken0` and `lpToken1`; the settlement `token` must match one of them.

Pool fee

`lpFee`. This DEX fee is separate from the platform fee.

Tick range

`lpTickLower` and `lpTickUpper`.

Accounted liquidity

`accountedLiquidity`; compare against `positions(lpTokenId)`.

Scope of accounted data

“Atlas liquidity” should mean assets associated with SC2's own LP position. It must not be replaced with the liquidity of the entire public pool, which also contains other liquidity providers' funds.

The Treasury balance, settlement-token reserve held by SC2, and third-party wallet balances are separate measures. They must not be added to the LP position without a separate label and a predefined methodology.

In the available code, `accountedLiquidity` reflects liquidity accounted for by the contract. External additions to the LP-NFT and DEX fees are not automatically counted as processed SC2 capital.

Verification procedure

1. Read every position parameter at the same block.
2. Reconcile tokens, fee, ticks and pool against Position Manager and factory data.
3. Calculate LP assets, the SC2 reserve and historical cash flows separately.

4. Ownership and Administrative Control

4.1 Fixed configuration

The addresses `token`, `treasury`, `insuranceWallet`, `manager`, and the parameters of the contract's own LP are set in the constructor. Some values are held in storage and others declared immutable; the supplied final code contains no external functions for arbitrary replacement of these addresses.

The supplied final code has no functions for arbitrary replacement of the LP position, transfer of the LP-NFT, issuance of external NFT approvals, or administrative withdrawal of funds. The contract uses the LP only through the accounting and payout operations provided for in the code. Reviewing source code does not replace an independent audit.

4.2 Manager permissions

Method	Permission
`schedulePolicy(change)`	Announce the policy for the next weekly period: monetary limits may be increased; tariff availability and the short-cycle share may be changed within the code's constraints.
`scheduleOpening(window, opensAt)`	Reschedule the opening time of a period that has not yet opened, within its calendar day.
`importSponsorBatch(...)` / `finalizeMigration()`	A one-time migration is available without the manager role, but checked against the count and hash fixed in the constructor. Completed migration cannot be repeated.

These permissions must not be confused with modifying executable code, arbitrarily withdrawing LP, or replacing fixed addresses.

4.3 Checking the controlling address

If `manager` is a Safe, record its code, threshold and signers. Old SC1 signer addresses and the previous Safe snapshot do not establish SC2's configuration. The manager's powers relate to admission of new cycles. In the supplied code, admission restrictions and tariff availability do not suspend claims for cycles already opened.

5. LP-NFT Ownership and Operator Approvals

Position Manager distinguishes the LP-NFT owner, approval for an individual NFT, and operator approval for all NFTs owned by an address. These three checks must be performed separately.

Check

Expected condition for the described model

`ownerOf(lpTokenId)`

SmartCycle2 address.

`getApproved(lpTokenId)`

Zero address.

`isApprovedForAll(SC2, operator)`

No third-party operators; check the code, approval history and relevant addresses.

`positions(lpTokenId)`

Tokens, fee and ticks match SC2 parameters.

How to repeat the check

1. Obtain `lpManager` and `lpTokenId` from SmartCycle2.
2. Open Read Contract for the specified Position Manager.
3. Read `ownerOf`, `getApproved` and `positions`.
4. Review `Approval` and `ApprovalForAll` events, then current permissions for the identified operators.
5. Compare `lpFactory` and the `getPool` result with `lpPool`.

What locking the LP means in plain language

The LP position is held by the smart contract. A participant or the administration cannot take it for themselves through a separate command. The contract uses liquidity for the calculations and payouts provided for by its code. Locking the position does not prohibit every movement of funds: regular cycle payouts and partner calculations use liquidity under the contract's rules.

6. Interactions and Flow of Funds

6.1 Smart Cycle flow

1. The participant verifies the SC2 address and grants it a settlement-token allowance.
2. A call to `create(tariffId, gross, sponsor)` creates a cycle after admission checks.
3. The contract records the cycle and structure data needed for calculations, accounting for incoming funds and the specified deductions.
4. Processed funds are placed in the contract's own LP position under SC2 rules.
5. Once the conditions are met, the owner calls `claim(id)`.
6. The contract uses SC2's accounted reserve. If it is insufficient, the shortfall is obtained from its own LP position. A successful Claim is recorded only after the whole required operation completes.

6.2 Partner reward flow

Under the agreed SC2 rules, the main bonus and matching are calculated using recorded data from the originating cycle and accumulated in the partner balance. The entire Lockup Flow bonus accrues when the cycle is opened. For Daily Flow, 20% accrues at opening; the remaining 80% is distributed over 200 days and accrues when the cycle owner receives payouts. Once 1 USDT is reached, the full available amount is transferred. A balance below 1 USDT remains until the threshold is reached.

`partnerBalance(address)` shows the recipient's accrued balance; `partnerReserved()` shows the amount reserved for such balances. `PartnerRewardAccrued` records accrual; `PartnerPayoutPaid` records a transfer to the recipient. When a new positive accrual brings the balance to 1 USDT, the contract transfers the entire balance. There is no separate manual partner-balance withdrawal method. `partnerSettlementState(id)` contains settlement markers for the originating cycle, not a wallet balance.

6.3 Simplified architecture

Source

Action / recipient

Participant

`create` / `claim` → SmartCycle2

SmartCycle2

Internal cycle accounting and partner calculations

SmartCycle2 ↔ Position Manager

Own LP-NFT ↔ registered pool

SmartCycle2

Specified payouts to participants and partners

SC2 incoming flow

Treasury and SC1 fund according to final-build rules

The diagram shows functional relationships. It does not mean that balances at all listed addresses belong to one liquidity metric.

7. Independent Verification Checklist

Contract identity

1. Compare every full address with the registry and verify the network.
2. Confirm that bytecode exists at the contract address.
3. Check source publication, compiler settings, constructor and correspondence with the final build.
4. Read `token`, `treasury`, `insuranceWallet`, `SYSTEM_SPONSOR`, `manager` and all LP parameters.

LP position and management

1. Check the LP-NFT owner, individual approval and operator approvals.
2. Compare the position's tokens, fee and tick range with SC2.
3. Check `manager`, `effectiveLimits`, `startWeek`, `scheduledWeek`, `followingWeek`, `policyAnnouncement` and `periodOpening`.
4. Check `migrationFinalized` and the one-time structure migration materials.

Funds and activity

1. Reconcile settlement-token transfers with `CycleOpened` and `CycleClaimed` events.
2. Compare `CycleOpened`, `CycleClaimed`, `PartnerRewardAccrued`, `PartnerPayoutPaid`, `ProcessedCapitalPlaced` and `ProcessedCapitalReturned` events with the final ABI.
3. Separate LP assets, reserves, incoming and outgoing flows, partner payments and fees.
4. For each snapshot, record the block, time, full addresses and transaction range used.

Independent audit

When the report is published, add the auditor, date, reviewed code version, scope, link to the full report and status of findings. Until then, this registry does not claim that SC2 has already been audited.

8. Fixed and Dynamic Values

Configuration recorded in the registry

1. SmartCycle2 address, final ABI version and network identifier.
2. Settlement token, Treasury, manager and the route for allocations to the SC1 fund confirmed in the final build.
3. Position Manager, factory, pool, LP-NFT, tokens, fee and ticks.
4. Identifier of the verified build and bytecode hash.
5. Snapshot block and time, and controlling-address status.

Values updated with every review

1. Current LP assets and their value at the selected block.
2. SC2 reserve, accounted liquidity and unprocessed inflows.
3. Number and status of cycles, available and completed payouts.
4. Incoming flow, outgoing flow, partner payments and fees.
5. Admission status, effective and announced limits.
6. Current approvals and external-contract status.

Period methodology

For daily, weekly, monthly or yearly statistics, specify the beginning and end of the period and corresponding block boundaries. Sum flows by transactions within the period. Show liquidity and balances at period end; do not sum them by day.

“Net flow” is calculated as incoming flow minus outgoing flow for the same set of addresses and period. Internal transfers between the contract's reserve and LP should not be counted again as user inflows or payouts.

The current value of the LP position is not the total liquidity of the public pool. None of these indicators alone proves solvency or guarantees future Claims.

9. Full Address and Link Index

The registry was updated on 19 October 2026 with the SC2 main contract, deployment wallet, limit management, recipients of the fee and fund allocation, and LP position No. 7682306. The project team supplied the stated purposes of the service addresses.

Purpose	Address / value
Smart Cycle 2	0x7E7BeDfcea906A1feF682ecb4e316F6bD87587E3
Deployment wallet	0x61726fbe5eC6b98d67dC4eA4B308FdFa0a18B424
Limit management	0xd587ba178010A34F7cFeEC6bd41D86eBEfAa11D3
Company fee, 6%	0x5F0411975A5500dBC465cceE16dEa1C7b93440f7
SC1 compensation fund, 2%	0x12aB84172C31DCE819aD2cbfcCFAD39C2D3d497f
Position Manager	0x46a15b0b27311cedf172ab29e4f4766fbe7f4364
LP-NFT / tokenId	7682306

Direct verification links

- SC2 contract on BscScan
- LP position on BscScan
- LP position on PancakeSwap

Scope of verification

The deployment wallet, limit management and recipients of funds have different roles. The 2% recipient was confirmed using `insuranceWallet()`. The system sponsor was confirmed using `SYSTEM_SPONSOR()`. These checks alone are not a security audit.

Confirmed network data

Check

Value at block 128510916

token

0x55d398326f99059ff775485246999027b3197955

treasury

0x5f0411975a5500dbc465ccee16dea1c7b93440f7

manager

0xd587ba178010a34f7cfeec6bd41d86ebefaa11d3

lpManager

0x46a15b0b27311cedf172ab29e4f4766fbe7f4364

lpFactory

0x0bfbcf9fa4f9c56b0f40a671ad40e0805a091865

lpPool

0x92b7807bf19b7dddf89b706143896d05228f3121

lpToken0

0x55d398326f99059ff775485246999027b3197955

lpToken1

0x8ac76a51cc950d9822d68b83fe1ad97b32cd580d

ownerOf

0x7e7bedfcea906a1fef682ecb4e316f6bd87587e3

getApproved

0x0000000000000000000000000000000000000000

lpTokenId

7682306

lpFee

100

lpTickLower

1000

lpTickUpper

2000

UTC time

2026-10-19T07:03:19+00:00

Verified by reading BNB Smart Chain at block 128510916. The LP-NFT belongs to the SC2 address; `getApproved` returns the zero address. This is a check of values at that snapshot, not a code audit or complete operator-approval analysis.

Additional final ABI cross-check

Method / block	Value
`insuranceWallet()`	`0x12aB84172C31DCE819aD2cbfcCFAD39C2D3d497f`
`SYSTEM_SPONSOR()`	`0x28498D22Ff08fc0309d0a82335E8960336003C6b`
Verification block	`128520088`

10. Registry Update Rules

Update the document when a new production build is published or its identifying data is corrected. Immutable parameters of the active contract must not be described as ordinary editable settings.

1. Record the source of each address and verify its full format.
2. Check code, constructor parameters and read-method values.
3. Record the new reference block and UTC time.
4. Update explorer, LP-NFT and audit links.
5. Check language versions, document button and archive links.
6. Retain the previous edition with its date and cycle attribution.

Document history

Date / edition	Change
22 July 2026 · 2.0	Original Atlas System registry and independent verification procedure.
25 July 2026 · 3.0	Expanded SC1 infrastructure, LP-NFT, Safe and operator-approval registry.
17 October 2026 · SC2	Full section structure retained. Description adapted to available SC2 sources; addresses and reference block left to be completed.
17 October 2026 · editorial review	Clarified reserve and LP, fund interface status, accrued versus transferred amounts, and limits of

checking against the available build.

19 October 2026 · SC2 addresses

Added supplied SC2 and service-wallet addresses, LP-NFT No. 7682306 and direct verification links.

The original 18-page SC1 PDF is retained separately without changes. This edition contains confirmed SC2 configuration data at the specified reference blocks and is not an independent audit report.

11. Important Explanation

The registry provides addresses, describes relationships and gives a reproducible verification method. It does not eliminate smart-contract, liquidity, market, operational-management, governance, key-custody or external-protocol risks.

Publication of source code on BscScan confirms that source files were matched with deployed bytecode as part of BscScan's verification procedure. Publication alone is not a security audit.

Participants and professional reviewers should obtain current data directly from BNB Chain, rather than rely only on screenshots, cached dashboards, rounded values or document copies without a version date and reference block number.

Basis for this edition

The complete Atlas System registry, version 3.0 of 25 July 2026: section structure and methodology retained. The description of current SC2 was refined using the supplied final SmartCycle2 sources and SmartCycle2Final.abi.json, including fees, partner balances and manager permissions.

Further configuration checks

Correspondence between the supplied sources and ABI and the executable SC2 bytecode has been confirmed. Permissions of external components and the actual LP state are checked separately. Confirmed reference values are given in Section 9. A link to the independent audit will be added once the report is published.

Edition: 19 October 2026 · atlas-system.global/documents/