

GENESIScoin Whitepaper v2.0

Public System Whitepaper

Version: v2.0 Public Release

Network: BNB Smart Chain

Token: GENESIScoin

Symbol: GENc

Project Identity: SQUNOHOLIX Brand Association — D_LAB

Core Standard: Proof Before Promise

Public Principle: Don't Assume. Verify.

1. Introduction

GENESIScoin is a modular on-chain system deployed on BNB Smart Chain.

It is not only a token contract.

GENESIScoin is built as a connected system of smart contracts, allocation modules, distribution mechanisms, automated trigger processes and long-term release structures. Each component has a defined role inside the ecosystem.

The system is designed around one core principle:

Proof Before Promise.

This means that GENESIScoin does not rely on theoretical claims as its foundation. The project prioritizes deployed infrastructure, visible contract state and verifiable execution before public narrative.

The GENESIScoin architecture separates the ecosystem into dedicated modules:

- the GENESIScoin token contract,
- the PreSaleGENc distribution module,
- the PublicSaleGENc distribution module,
- Asset Manager v1 for fixed-schedule treasury releases,
- Asset Manager v2 for conditional operational execution,
- RM01 as a separate strategic release module,
- defined liquidity allocations for LP Main and LP Backup,
- and a future LP Manager / liquidity-governance smart-contract layer.

The purpose of this whitepaper is to present the current GENESIScoin system as it exists now: its architecture, allocation model, deployed modules, execution logic and development standard.

This document is not a technical audit.

It is not a legal document.

It is not a promise of future outcome.

It is a public system overview.

The source of truth is the deployed system and the public blockchain state.

2. Proof Before Promise

GENESIScoin follows the Proof Before Promise development standard.

This standard is simple:

Do not ask people to believe what can be verified.

In Web3, claims are easy to make. Execution is harder to prove. GENESIScoin is structured around the opposite order:

1. build the system,
2. deploy the contracts,
3. define the rules,
4. expose the structure,
5. allow others to verify it.

The system is designed so that critical rules are not presented only as written promises. They are implemented through deployed contracts, visible allocation structures, release schedules and transaction-triggered execution paths.

This creates a different standard for project communication.

GENESIScoin does not use Proof Before Promise as a slogan. It is applied as a system design rule:

- allocations should be visible,
- modules should be identifiable,
- contracts should define execution,

- automation should trigger only what the contracts allow,
- public claims should match verifiable state.

This standard also connects directly to DAV:

Don't Assume. Verify.

GENESIScoin is built for users, partners and external observers who want to inspect what exists instead of relying only on what is said.

Proof Before Promise is the foundation.

DAV is the behavior.

3. GENESIScoin Overview

GENESIScoin is the base asset and system token of the GENESIScoin ecosystem.

It operates on BNB Smart Chain as a BEP-20 compatible token and connects to multiple system modules responsible for distribution, reserves, rewards, vesting, conditional operational execution and liquidity planning.

The system is designed as a modular architecture.

Instead of placing all logic inside one contract, GENESIScoin separates responsibilities across specialized contracts and execution layers. This makes each part of the system easier to understand, inspect and verify.

The current architecture includes:

- a core token layer,
- a staged PreSale layer,
- a PublicSale and distribution layer,
- a treasury vesting layer,
- a conditional operational execution layer,
- a strategic release module,
- and a liquidity allocation layer.

Each module serves a different function.

The token is the asset layer.

PreSale and PublicSale manage distribution.

Asset Manager v1 controls fixed-schedule treasury release.

Asset Manager v2 controls conditional operational execution and routing.

RM01 controls a separate strategic release schedule.

LP Main and LP Backup define liquidity allocations.

Automation processes trigger transactions when system conditions are reached.

The key model is:

contracts execute, bots trigger, operators activate.

This means that off-chain automation does not replace smart-contract logic. Bots and scripts observe conditions, submit transactions and support execution timing. The contracts themselves define what can happen and whether a transaction is valid.

4. Token Overview

4.1 Token Identity

Field	Value
Token name	GENESIScoin
Symbol	GENc
Network	BNB Smart Chain
Standard	BEP-20 / ERC-20 compatible
Decimals	18
Token contract	0x8d9f95Dd624F581803e06652623897DfeCB82CA6

GENc is the base token used across the GENESIScoin system.

It is used inside the PreSale module, PublicSale module, treasury release modules, conditional operational execution modules, reward mechanisms, airdrop systems and liquidity allocation structure.

4.2 Supply Model

The original GENESIScoin allocation model was built around a total allocation of:

100,000,000,000 GENc

This original allocation defines the intended structure of the ecosystem, including:

- PreSale distribution,
- B100D loyalty allocation,
- PublicSale bonus allocation,
- DividendGENc pool,
- treasury reserves,
- operational allocations,
- partner allocations,
- airdrop allocations,
- and liquidity allocations.

GENESIScoin supply originates from deployment-time allocation. No external post-deployment minting path is presented in this whitepaper unless verified otherwise from the deployed source.

The final public version of this whitepaper should always use the current on-chain total supply at the time of publication.

4.3 Token Function Inside the System

GENc is not isolated from the rest of the architecture.

It connects to:

- PreSaleGENc for staged token distribution,
- PublicSaleGENc for post-PreSale distribution mechanisms,
- Asset Manager v1 for treasury release schedules,
- Asset Manager v2 for conditional operational execution,
- RM01 for strategic scheduled release,
- tracker and verifier systems for distribution eligibility,
- and liquidity allocations connected to market activation.

GENc acts as the system asset layer that all other modules operate around.

5. Token Fee and Trading Activation Model

GENESIScoin includes token-level configuration surfaces connected to trading activation, AMM settings, fee handling, fee exclusions and dividend exclusions.

These surfaces are part of the token control layer and should be understood separately from the PreSale, PublicSale, Asset Manager and release-module logic.

The token contract may define or expose configuration paths related to:

- trading activation,
- automated market maker settings,
- transaction-fee handling,
- fee exclusions,
- dividend exclusions,
- reward-related fee configuration,
- liquidity-related fee configuration,
- burn-related fee configuration,
- and operational recipient configuration.

The exact active configuration must always be verified against the deployed token contract before publication, integration or external analysis.

Trading activation is also a separate state from liquidity allocation.

LP Main and LP Backup define liquidity allocations. They do not, by themselves, prove that active liquidity has been deployed or that a live trading pair is active.

Token-level trading activation, AMM configuration and liquidity deployment are connected, but they are not the same thing.

The final documentation should not present a historical fee table as current active configuration unless it has been checked against the deployed token source and current chain state.

6. Modular System Architecture

GENESIScoin is structured as a modular system.

Each module has a defined role and a separate responsibility. This separation makes the system easier to inspect because no single public explanation needs to hide different mechanisms under one generic label.

The architecture is composed of the following core components.

6.1 GENESIScoin Token Contract

The GENESIScoin token contract is the base asset layer of the ecosystem.

It defines the GENc token and provides the transfer layer used by every other module.

Core role:

- token accounting,
- transfer execution,
- token-level configuration surfaces,
- trading activation path,
- AMM configuration surfaces,
- fee and exclusion control surfaces,
- integration with distribution and reward mechanisms.

The token contract is not the full system by itself.

It is the asset layer that the rest of the GENESIScoin architecture operates around.

The token contract may also hold a changing operational GENc balance connected to transfer, buy, sell or fee-related routing mechanics. That balance should not be interpreted as a fixed allocation pool.

6.2 PreSaleGENc

PreSaleGENc is the staged initial distribution module.

It manages the PreSale process and connects several early-system mechanics, including stage progression, whitelist timing and B100D entry.

Core role:

- staged PreSale structure,
- stage progression,
- purchase limits,
- BNB price update path,
- whitelist opening and closing,
- B100D registration,
- B100D daily payout logic,
- final bonus trigger path,
- BNB proceeds distribution.

PreSaleGENc is contract-executed and oracle-supported.

The contract defines the rules.

The Oracle triggers selected transitions.

Bots support timing and participant-state enforcement.

PreSaleGENc is also an important timing anchor for downstream modules because whitelist closure is used by later execution layers.

6.3 PublicSaleGENc

PublicSaleGENc is the post-PreSale distribution module.

It becomes relevant after the required PreSale and whitelist conditions are satisfied.

Core role:

- PublicSale activation,
- daily bonus execution,
- PublicSale bonus pool custody,
- DividendGENc pool custody,
- airdrop pool custody,
- list-based airdrop execution,
- adjusted GENc distribution execution.

PublicSaleGENc holds and transfers post-PreSale distribution pools.

Eligibility observation and list preparation may happen outside the contract through dedicated scripts, but final token transfers are executed by the contract.

This means PublicSaleGENc is the execution layer for multiple post-PreSale distribution mechanisms.

6.4 Asset Manager v1

Asset Manager v1 is the fixed-schedule treasury release module.

The deployed contract name is AssetManager.

It controls long-term treasury and ecosystem reserve allocations through predefined release schedules.

Core role:

- Team allocation release schedule,
- Marketing allocation release schedule,
- Development allocation release schedule,
- Event / Expo allocation release schedule,
- Future Project allocation release schedule,
- SBA reserve release schedule.

Asset Manager v1 follows a time-based model.

Its release logic is anchored to WasteStart, which is set after the PreSale whitelist close condition.

Once WasteStart is set, releases follow predefined schedules and destination paths.

Asset Manager v1 should be understood as the deterministic treasury release layer.

It does not operate as a dynamic operational controller.

6.5 Asset Manager v2

Asset Manager v2 is the conditional operational execution controller.

The deployed contract name is AssetManagerV2.

It is separate from Asset Manager v1.

Asset Manager v1 controls fixed treasury release schedules.

Asset Manager v2 controls dynamic operational execution paths.

Core role:

- Stage Airdrop Pool execution,
- stage PASS routing,
- stage FAIL routing,
- Milestone Pool execution,
- Partners Pool execution,
- Contest / Promotional Pool execution.

Asset Manager v2 is not a passive reserve wallet.

It controls how selected GENc pools can move depending on system outcome, approved operations and contract-defined conditions.

For stage-based execution:

- a successful stage routes the defined stage allocation to the airdrop wallet,
- a failed eligible stage routes the defined stage allocation through the fail path,
- the fail path redirects GENc into burn and DividendGENc-related technical distribution routes.

This makes Asset Manager v2 a conditional routing layer.

Milestone, partner and contest-related operations use proposal, approval and execution paths where required.

The full AMv2 mechanics are described separately in the Asset Manager v2 section of this whitepaper.

6.6 RM01

RM01 is a separate deployed strategic release module.

It is not Asset Manager v1.

It is not Asset Manager v2.

It is not LP Main.

It is not LP Backup.

It is not the LP Manager.

RM01 controls a dedicated 50,000,000 GENc strategic release schedule.

Core role:

- sealed strategic pool,
- TGE trigger,
- scheduled release execution,
- hardcoded recipient path,
- RM_BOT-triggered execution.

RM01 is deployed as its own contract.

It was funded from the Partners wallet / Partners allocation path and sealed after full funding.

Allocation-wise, RM01 belongs to the RM01–RM10 reserve inside the 3,000,000,000 GENc Partners allocation. Contract-wise, RM01 is separate from Asset Manager v2. This distinction prevents double-counting and prevents RM01 from being incorrectly described as an AMv2 internal pool.

6.7 Liquidity Allocations

GENESIScoin includes two defined liquidity allocations:

- LP Main — 10,000,000,000 GENc,
- LP Backup — 10,000,000,000 GENc.

Together, these represent the liquidity allocation layer of the original system model.

LP Main and LP Backup are allocation wallets.

They are not the same as:

- an active LP pair,
- a locked liquidity contract,
- an LP Manager contract,
- Asset Manager v1,
- Asset Manager v2,
- RM01.

The LP Manager remains a future liquidity-governance smart-contract module until it is deployed and funded on-chain. Until then, LP Main and LP Backup must be described as liquidity allocation wallets.

Active liquidity deployment must be verified through actual on-chain liquidity transactions and pair state.

6.8 Automation Layer

GENESIScoin uses automation scripts and bots to support execution timing.

These processes do not rewrite contract rules.

They:

- observe events,
- read contract state,
- check timing conditions,
- submit transactions,
- and support staged execution.

contracts execute, bots trigger, operators activate.

Some bots are expected to run during broad system phases.

Some bots are phase-driven and activated before they are needed.

Some scripts are manual CLI tools and are not persistent daemons.

This distinction matters because runtime state does not define the whole system.

A stopped bot outside its activation window does not mean a function does not exist.

It means that process is not active in that phase.

6.9 Architecture Summary

GENESIScoin is designed as a system of connected but separated modules.

Layer	Module	Main role
Asset layer	GENESIScoin Token	Base GENc token
Initial distribution	PreSaleGENc	Staged PreSale and B100D
Public distribution	PublicSaleGENc	PublicSale, bonus, DividendGENc and airdrop pools
Fixed treasury release	Asset Manager v1	ID1–ID6 fixed-schedule releases
Conditional operational execution	Asset Manager v2	Stage, milestone, partners and contest routing
Strategic release	RM01	First 50M GENc RM release module
Liquidity allocation	LP Main / LP Backup	Liquidity reserve wallets
Future liquidity governance	LP Manager	Future liquidity-governance contract layer
Automation	Bots / scripts	Trigger layer for contract-defined actions

This modular structure allows GENESIScoin to separate long-term allocation, operational distribution, public participation, strategic releases, treasury vesting and liquidity planning into distinct system components.

The system standard remains:

Proof Before Promise.
Don't Assume. Verify.

7. Original Allocation Model

The original GENESIScoin allocation model was structured around a total allocation of:

100,000,000,000 GENc

This allocation defines the original tokenomics structure of the system.

It separates GENc supply into public distribution, holder distribution systems, treasury reserves, ecosystem reserves, partner allocation and liquidity allocation.

The original allocation model should not be confused with current custody.

View	Meaning
Original allocation	What each supply segment was intended for
Current custody	Where GENc currently sits
Deployed module	Which contract or wallet controls a system path
Release logic	How an allocation moves over time

Both views matter.

The original allocation explains purpose.
Current custody explains present location.
Execution logic explains movement.

7.1 Original GENc Allocation

Allocation	Amount	Share
Pre-Sale + Bonus	35,800,000,000 GENc	35.8%
B100D Loyalty Pool	3,000,000,000 GENc	3.0%
DividendGENc Pool	10,000,000,000 GENc	10.0%
Public Sale Bonus Pool	3,000,000,000 GENc	3.0%
DevG	1,000,000,000 GENc	1.0%
Partners	3,000,000,000 GENc	3.0%
Event / Expo	3,000,000,000 GENc	3.0%
Airdrop	5,000,000,000 GENc	5.0%
Marketing	5,000,000,000 GENc	5.0%
TeamG	5,000,000,000 GENc	5.0%
Future Project	6,200,000,000 GENc	6.2%
LP Backup	10,000,000,000 GENc	10.0%
LP Main	10,000,000,000 GENc	10.0%

Total: 100,000,000,000 GENc

7.2 Allocation by System Purpose

The original allocation model can be grouped into five functional categories:

Category	Allocation areas
Public distribution	Pre-Sale + Bonus, Public Sale Bonus Pool, Airdrop
Holder distribution systems	B100D Loyalty Pool, DividendGENc Pool
Treasury and ecosystem reserves	DevG, Marketing, TeamG, Event / Expo, Future Project
Partner and operational allocation	Partners allocation, partner-related execution paths, RM01-RM10 reserve
Liquidity allocation	LP Main, LP Backup

This purpose-based view explains why each allocation exists.

7.2.1 Public Distribution

Public distribution includes allocations used for direct ecosystem participation and token distribution.

Allocation	Purpose
Pre-Sale + Bonus	Initial staged distribution

Allocation	Purpose
Public Sale Bonus Pool	Daily bonus mechanism connected to PublicSale
Airdrop	Holder, campaign and ecosystem distribution

These allocations are connected to participant distribution mechanics.

They are not treasury allocations.

7.2.2 Holder Distribution Systems

Holder distribution systems are technical GENc distribution mechanisms.

Allocation	Purpose
B100D Loyalty Pool	Daily and final bonus distribution for eligible PreSale-connected participants
DividendGENc Pool	Cycle-based GENc token distribution from a dedicated pool

DividendGENc is a technical system label for GENc token distribution.

It should not be interpreted as:

- a legal dividend,
- ownership right,
- profit share,
- yield product,
- income promise.

B100D is also a technical loyalty mechanism.

It is based on registration, initialHold tracking and continued holding behavior.

7.2.3 Treasury and Ecosystem Reserves

Treasury and ecosystem reserves support long-term development, operations and ecosystem continuity.

Allocation	Purpose
DevG	Development reserve
Marketing	Marketing reserve
TeamG	Team allocation
Event / Expo	Event and ecosystem promotion reserve
Future Project	Future project development and SBA reserve structure

In the current modular architecture, the 6,200,000,000 GENc Future Project allocation is structurally split into:

- Future Project Development,
- SBA ecosystem reserve.

These allocations are controlled through the fixed-schedule treasury release layer.

The relevant deployed module for this category is:

Asset Manager v1 / AssetManager

Asset Manager v1 controls the ID1-ID6 fixed-schedule release model.

7.2.4 Partner and Operational Allocation

The Partners allocation is:

3,000,000,000 GENc

This allocation supports partner-related ecosystem paths and strategic release structures.

In the deployed system, partner-related execution must be separated into three different layers:

Layer	Meaning
Partners allocation	3B GENc original tokenomics allocation
AMv2 Partners Pool execution path	Contract-controlled partner-related operational route
RM01-RM10 reserve	500M GENc strategic release reserve inside the Partners allocation

These layers are connected, but they are not the same thing.

Asset Manager v2 may route partner-related GENc through its Partners Pool execution path to the Partners wallet.

RM modules are separate release contracts.

RM modules are not internal AMv2 pools.

RM01-RM10 Reserve

Inside the 3,000,000,000 GENc Partners allocation, **500,000,000 GENc** is reserved for the RM01-RM10 strategic release module sequence.

Each RM module is structured around **50,000,000 GENc**.

RM Layer	Amount	Status
RM01	50,000,000 GENc	Deployed / verified / funded / sealed
RM02-RM10	450,000,000 GENc	Reserved for future RM modules
Total RM01-RM10 reserve	500,000,000 GENc	Sub-allocation inside Partners allocation

This means RM01-RM10 are not additional allocation outside the 3B Partners allocation.

They are a reserved strategic release layer inside the Partners allocation.

The remaining Partners allocation after the RM01-RM10 reserve is:

2,500,000,000 GENc

The RM01-RM10 reserve is a tokenomics-level sub-allocation.

RM01 is not extra supply.

RM01 is the first deployed release module from the RM01-RM10 reserve inside the Partners allocation.

7.2.5 Liquidity Allocation

GENESIScoin includes two dedicated liquidity allocations.

Allocation	Amount	Purpose
LP Main	10,000,000,000 GENc	Primary liquidity allocation
LP Backup	10,000,000,000 GENc	Backup / secondary liquidity allocation

Together, LP Main and LP Backup form the liquidity allocation layer.

At the current stage, they must be described as liquidity allocation wallets unless and until a deployed LP Manager or another verified liquidity-governance structure takes custody of those balances on-chain.

A liquidity allocation is not the same as active liquidity.

Active liquidity must be verified through actual liquidity transactions and pair state.

7.3 Deployment-Aware Allocation View

As the system evolved, the original allocation model was mapped into deployed modules and defined custody structures.

Allocation Area	Current System Layer
Pre-Sale + Bonus	PreSaleGENc
B100D Loyalty Pool	PreSaleGENc
Public Sale Bonus Pool	PublicSaleGENc
DividendGENc Pool	PublicSaleGENc
Airdrop execution paths	PublicSaleGENc / Asset Manager v2 / tracker layer
TeamG	Asset Manager v1
Marketing	Asset Manager v1
DevG	Asset Manager v1
Event / Expo	Asset Manager v1
Future Project	Asset Manager v1
SBA Reserve	Asset Manager v1
Stage Airdrop Pool	Asset Manager v2
Milestone Pool	Asset Manager v2
Partners Pool execution path	Asset Manager v2 / Partners wallet path
Contest / Promotional Pool	Asset Manager v2
Partners allocation	3B GENc tokenomics allocation
RM01-RM10 strategic release reserve	500M GENc sub-allocation inside the 3B Partners allocation
RM01 strategic release module	First deployed 50M GENc module from the RM01-RM10 reserve
LP Main	Liquidity allocation wallet
LP Backup	Liquidity allocation wallet
LP Manager	Future liquidity-governance contract layer

RM01 is documented as a separate deployed release module because it has its own contract, pool, seal state, recipient path and RM_BOT-triggered schedule.

Allocation-wise, RM01 belongs to the RM01–RM10 reserve inside the 3B GENc Partners allocation.

Contract-wise, RM01 is separate from Asset Manager v2.

It must not be counted as additional supply outside the original allocation model.

7.4 Allocation vs Custody

The original allocation model and current custody view are different layers.

View	Meaning
Original allocation	Intended tokenomics structure
Current custody	Where GENc currently sits
Deployed module	Contract that controls or executes a system path
Allocation wallet	Wallet holding an allocation outside a deployed module
Release logic	How an allocation moves over time
Automation	How contract-defined actions are triggered

This distinction matters because a current holder balance does not automatically explain the full tokenomics model.

The final GENESIScoin documentation must show:

- what was allocated,
- where it currently sits,
- what contract or wallet controls it,
- and what logic governs movement.

7.5 Allocation Documentation Rule

The final GENESIScoin documentation must keep the following rule clear:

Original allocation is the tokenomics model.

Current custody is the deployed or wallet-held state.

Release logic is how allocation moves.

Operational automation is how contract-defined actions are triggered.

This prevents one of the most common documentation errors: presenting an original allocation table as if it also proves current custody.

GENESIScoin v2.0 documentation must present both layers clearly.

8. Current Deployed System State

GENESIScoin has moved beyond a single-token model into a deployed modular system.

The current system includes deployed contracts, allocation wallets, execution paths and operational balances. These layers must be separated because they do not mean the same thing.

This section explains the current deployed state of the system through four views:

- deployed modules,
- token contract behavior,
- visible custody snapshot,
- execution interpretation.

A deployed module shows system architecture.

A custody snapshot shows where GENc balances are currently visible.

An allocation model explains intended purpose.

Execution logic explains how tokens move.

These views must not be merged.

8.1 Deployed Core Modules

Module	Address	Status	Role
GENESIScoin Token	0x8d9f95Dd624F581803e06652623897DfeCB82CA6	Deployed	Base token contract and asset layer
PreSaleGENc	0x019A2D76D825914B5b52552e1A1D09F024CC4C58	Deployed / verified	Staged PreSale execution module

Module	Address	Status	Role
PublicSaleGENc	0x74eb74d66e8bd29226228536307BF5588168A8b7	Deployed / verified	Post-PreSale distribution module
Asset Manager v1 / AssetMenager	0x673eC5B98bC6d8A3b633F8D900B4C0832850423D	Deployed / verified	Fixed-schedule treasury release module
Asset Manager v2 / AssetManagerV2	0xA0e6e93cFc4507Fc2687FA3aA943F238C28Bee97	Deployed / verified	Conditional operational execution controller
RM01	0x40A830b22E4C39D16B8b17e8beD4ED0f7C29E547	Deployed / verified / funded / sealed	Strategic release module

The GENESIScoin token contract is the base asset layer.

PreSaleGENc, PublicSaleGENc, Asset Manager v1, Asset Manager v2 and RM01 are deployed system modules.

LP Main and LP Backup are not listed as deployed modules because they are liquidity allocation wallets. They are shown separately in the custody snapshot.

8.2 GENESIScoin Token Contract Balance Behavior

The GENESIScoin token contract address may hold a variable GENc balance.

This balance should not be interpreted as a fixed allocation pool.

The token contract balance can change because of token-level mechanics such as:

- transfer-related routing,
- internal token flow,
- fee accumulation if enabled,
- temporary holding before later routing,
- operational token movement connected to transfer, buy or sell logic.

At the current stage, buy and sell paths may not yet be active, while transfer-related mechanics can still affect the token contract balance.

This means the GENESIScoin token contract balance is dynamic and operational.

It is not:

- a treasury pool,
- a static allocation,
- a locked reserve,
- or a separate tokenomics category.

The token contract is therefore shown as a deployed module, not as a fixed custody allocation.

8.3 Current Visible Custody Snapshot

The holder view shows where major GENc balances are visible at the time of observation.

This is a custody snapshot.

It does not replace the original allocation model.

Holder / Allocation	Address / Label	Current visible GENc balance	Classification
PreSaleGENc	0x019A2D76D825914B5b52552e1A1D09F024CC4C58	38,793,884,953.9616 GENc	Deployed contract custody
Asset Manager v1 / AssetMenager	0x673eC5B98bC6d8A3b633F8D900B4C0832850423D	20,200,000,000 GENc	Deployed contract custody
PublicSaleGENc	0x74eb74d66e8bd29226228536307BF5588168A8b7	13,100,000,000 GENc	Deployed contract custody
LP Backup	0x5807804228dc9AbD8b6FFddA7C5226CBe5AD9d86	10,000,000,000 GENc	Liquidity allocation wallet
LP Main	0xc627BDd1B91E69981E7b59635d0154C0678A6991	10,000,000,000 GENc	Liquidity allocation wallet
Asset Manager v2 / AssetManagerV2	0xA0e6e93cFc4507Fc2687FA3aA943F238C28Bee97	7,630,000,000 GENc	Deployed contract custody
RM01	0x40A830b22E4C39D16B8b17e8beD4ED0f7C29E547	50,000,000 GENc	Deployed / sealed strategic release module
Airdrop wallet	Airdrop GENc	30,938,500 GENc	Airdrop allocation wallet

This table separates deployed contract custody from allocation-wallet custody.

LP Main and LP Backup are liquidity allocation wallets. They are not active LP pairs and are not LP Manager custody.

RM01 is separate at contract level, but allocation-wise it belongs to the RM01–RM10 reserve inside the 3B GENc Partners allocation.

The custody snapshot should be refreshed before final publication.

8.4 Current Module Roles

GENESIScoin Token

The GENESIScoin token is the base asset layer of the system.

It provides the GENc token used by all distribution, release, reward, airdrop and liquidity-related modules.

The token contract may also hold a changing operational balance connected to token-level routing mechanics. This balance is dynamic and should not be treated as a fixed allocation.

PreSaleGENc

PreSaleGENc is the staged initial distribution module.

It controls the PreSale flow, stage progression, whitelist logic, B100D entry path, B100D payout logic and PreSale-related fund distribution mechanics.

The current visible holder balance shows that PreSaleGENc remains the largest deployed contract custody module.

PublicSaleGENc

PublicSaleGENc is the post-PreSale distribution module.

It controls PublicSale activation and executes daily bonus, DividendGENc and airdrop-related transfers.

PublicSaleGENc is a deployed contract custody layer for post-PreSale distribution pools.

Asset Manager v1 / AssetManager

Asset Manager v1 controls fixed-schedule treasury allocations.

It manages the ID1-ID6 release layer for long-term treasury and ecosystem reserves.

Its logic is time-based.

Once the release anchor is set, releases follow predefined schedules and destination paths.

Asset Manager v2 / AssetManagerV2

Asset Manager v2 is the conditional operational execution controller.

It is more advanced than a passive reserve wallet.

AMv2 controls conditional and approval-based operational flows, including:

- stage-based release or fail logic,
- milestone-based releases,
- partner-related execution paths,
- contest and promotional execution paths.

The key distinction is that AMv2 does not only hold GENc.

It determines how GENc is routed depending on system outcome or approved operation.

For stage-based execution:

- a successful stage routes the defined stage amount to the airdrop wallet,
- a failed eligible stage routes the defined stage amount through the fail path.

The fail path is structurally different from a normal release. It does not simply keep the tokens in the pool. It redirects the stage amount into burn and DividendGENc-related distribution paths according to the contract-defined logic.

This makes AMv2 a conditional routing layer, not a simple operational wallet.

A full AMv2 mechanics section is documented separately in the Asset Manager v2 section.

RM01

RM01 is the first deployed strategic release module from the RM01-RM10 reserve.

It controls a 50,000,000 GENc sealed pool, funded from the Partners allocation or Partners wallet path and released according to its own contract-defined schedule.

RM01 belongs allocation-wise to the 500,000,000 GENc RM01-RM10 reserve inside the 3,000,000,000 GENc Partners allocation.

RM01 is documented separately because it is deployed as its own release module.

RM01 is not Asset Manager v2.

Asset Manager v2 may control partner-related operational flows, but RM01 executes its own separate release schedule through its own contract.

LP Main and LP Backup

LP Main and LP Backup are liquidity allocation wallets.

They are not:

- active LP pairs,
- LP Manager custody,
- locked liquidity contracts,
- Asset Manager v1,
- Asset Manager v2,
- RM01.

They represent reserved liquidity supply until active liquidity deployment or LP Manager custody can be verified on-chain.

8.5 Current System Statement

The current GENESIScoin system consists of:

- a deployed token layer,
- a deployed PreSale module,
- a deployed PublicSale module,
- a deployed fixed treasury release module,
- a deployed conditional operational execution controller,
- a deployed RM01 strategic release module,
- defined LP Main and LP Backup liquidity allocation wallets,
- an airdrop allocation wallet,
- and a future LP Manager / liquidity-governance smart-contract layer.

This creates a modular system where each major function is separated:

Area	Current layer
Base asset	GENESIScoin Token
Initial staged distribution	PreSaleGENc
Post-PreSale distribution	PublicSaleGENc
Fixed treasury releases	Asset Manager v1
Conditional operational routing	Asset Manager v2
Strategic scheduled release	RM01
Liquidity allocation	LP Main / LP Backup
Future liquidity governance	LP Manager / liquidity-governance smart-contract layer

The system should be read through both architecture and custody.

Architecture shows what exists.

Custody shows where visible balances currently sit.

Execution logic explains how balances can move.

8.6 Interpretation Rule

The final documentation must preserve the following distinction:

View	Meaning
Deployed modules	Shows the system architecture
Visible custody snapshot	Shows where major GENc balances currently sit
Original allocation model	Shows intended tokenomics purpose
Release mechanics	Shows how contract-held allocations move
Operational automation	Shows how contract-defined actions are triggered

This prevents a major documentation error: treating a visible holder balance as if it were the entire allocation model.

GENESIScoin v2.0 documentation should therefore separate:

- what was allocated,
 - where it currently sits,
 - which contract or wallet holds it,
 - and what logic controls future movement.
-

8.7 Publication Check Requirement

Before the final public release of this whitepaper, the following values must be checked against current on-chain state:

1. current total supply,
2. current GENESIScoin token contract balance,
3. current PreSaleGENc balance,
4. current PublicSaleGENc balance,
5. current Asset Manager v1 balance,
6. current Asset Manager v2 balance,
7. current RM01 balance and release state,
8. current LP Main balance,
9. current LP Backup balance,
10. current airdrop wallet balance,
11. PublicSale activation state,
12. active LP pair status,
13. LP Manager deployment status,
14. RM01–RM10 reserve reconciliation inside the 3B GENc Partners allocation.

This whitepaper should always align with the current blockchain state at the time of publication.

9. PreSale System

PreSaleGENc is the staged initial distribution module of the GENESIScoin system.

It defines the first major distribution path and connects multiple downstream system events, including Stage progression, B100D whitelist registration, B100D payout logic, PublicSale preparation and release-module anchors.

The PreSale system is not only a purchase mechanism.

It is one of the main execution anchors of the GENESIScoin architecture.

9.1 PreSale Purpose

The PreSale system is designed to:

- distribute GENc through defined stages,
- control early participation limits,
- apply stage-specific logic,
- connect Stage 5 to B100D whitelist registration,
- establish the PreSale end state,
- trigger the whitelist close window,
- support B100D activation,
- support PublicSale activation,
- and provide timing anchors for downstream system modules.

9.2 Stage Model

The PreSale model is organized into five stages.

Each stage exists as part of a larger system sequence.

Stage	Function
Stage 1	Initial threshold stage
Stage 2	Timed or sold-out progression stage
Stage 3	Timed or sold-out progression stage
Stage 4	Timed or sold-out progression stage
Stage 5	Final PreSale stage and B100D whitelist entry point

9.3 Stage 1 — Initial Threshold

Stage 1 is the first major execution threshold.

When 50% of Stage 1 is sold, the liquidity/trading activation path becomes relevant under the GENESIScoin staged model.

This does not mean that public trading must activate immediately at that moment.

It means that Stage 1 creates the first major system threshold connected to later liquidity and trading activation.

Stage 1 50% sold is a structural liquidity/trading activation threshold.

9.4 Stages 2-4 — Timed or Sold-Out Progression

Stages 2-4 follow a bounded progression model.

Each stage can progress through:

1. sold-out completion, or
2. maximum duration expiry.

The maximum duration model prevents these stages from remaining open indefinitely.

The Oracle monitors active stages and can close a stage when the configured duration has expired.

This creates a progression structure where Stages 2-4 are limited by either demand or time.

Stages 2-4 are bounded stages: they can advance through sell-out or by reaching their maximum duration.

9.5 Stage 5 — Final PreSale Stage and Whitelist Open

Stage 5 is the final PreSale stage.

When Stage 5 begins, the B100D whitelist registration path opens through the Oracle-triggered whitelist function.

This creates a direct connection between the final PreSale phase and the B100D loyalty system.

Stage 5 does not mean B100D has started.

It means:

the B100D whitelist registration window is open.

B100D starts later, after PreSale completion, whitelist closure and the required delay.

9.6 PreSale End and Whitelist Close

When the PreSale reaches its ending condition, the system records the PreSale end state.

After PreSale end, a 24-hour window leads into whitelist closure.

The whitelist close event is a major system anchor.

It affects:

- B100D start preparation,
- PublicSale activation path,
- Asset Manager v1 WasteStart anchor,
- RM01 TGE reference,
- post-PreSale tracking modes.

Whitelist close is not only the end of registration.

It becomes a structural timestamp used by multiple downstream modules.

9.7 B100D Connection

B100D is integrated with the PreSale system.

The B100D flow is:

1. Stage 5 begins,
2. whitelist registration opens,
3. eligible users register,
4. PreSale ends,
5. 24h whitelist close window completes,
6. whitelist closes,
7. required delay passes,
8. B100D starts,
9. daily B100D payouts begin,
10. participant holding condition is monitored,
11. final bonus path becomes available after B100D completion.

B100D is contract-executed and bot-supported.

The contract records participant state and executes payout paths.

The B100D bot observes balances and can trigger ejection when a participant's balance falls below the required initialHold.

9.8 Oracle Role in PreSale

The Oracle is the main trigger layer for PreSale transitions.

It supports:

- BNB price updates,
- stage duration checks,
- stage closure,
- whitelist opening,
- whitelist closing,
- B100D start,
- daily B100D payout triggers,
- BNB proceeds distribution,
- PublicSale start.

The Oracle does not replace the contract.

The Oracle submits transactions to functions that the contract allows.

The contract defines the rules. The Oracle triggers selected transitions.

9.9 BNB Distribution

The PreSale system includes contract-defined BNB routing logic.

BNB raised through the PreSale is distributed through predefined operational categories.

The historical model includes routing to:

- liquidity,
- team,
- development,
- marketing,
- community,
- SecureAudit,
- LevelUpVision,
- project expansion.

Before final publication, the current deployed PreSale source and current recipient addresses must be checked to confirm the exact routing table.

BNB routing follows contract-defined distribution logic.

9.10 PreSale Summary

PreSaleGENc is the first execution engine of the GENESIScoin system.

It does more than process purchases.

It creates the staged entry model, opens the B100D registration path, defines the PreSale end condition, anchors whitelist closure and supports downstream execution for B100D, PublicSale and release modules.

PreSale is the first major proof layer of the GENESIScoin architecture.

10. PublicSale System

PublicSaleGENc is the post-PreSale distribution module of the GENESIScoin system.

It becomes relevant after PreSale and whitelist conditions are satisfied.

The PublicSale module is responsible for distribution mechanics that operate after the initial PreSale phase, including PublicSale activation, daily bonus execution, DividendGENc pool distribution and airdrop execution.

10.1 PublicSale Purpose

PublicSaleGENc is designed to:

- activate the public distribution phase,
- execute daily bonus transfers,
- hold PublicSale bonus pool balances,
- hold DividendGENc pool balances,
- hold airdrop pool balances,
- execute list-based airdrop transfers,
- execute adjusted GENc distribution payouts,
- and support post-PreSale participation mechanics.

The contract is the custody and execution layer.

External automation supports observation, eligibility tracking and trigger timing.

10.2 PublicSale Activation

PublicSale activation occurs after the required PreSale and whitelist conditions have been reached.

The Oracle checks the relevant state and triggers the PublicSale start function when the activation condition is satisfied.

This creates an on-chain PublicSale start timestamp.

PublicSale is activated through the Oracle trigger after the required PreSale and whitelist conditions are satisfied.

10.3 Daily Bonus System

The PublicSale daily bonus system is connected to qualifying market activity.

The eligible transaction flow is:

1. PublicSale is active,
2. a token transfer is observed from the configured LP pair,
3. eligibleBot checks the current PublicSale day and phase,
4. eligibleBot checks the minimum buy requirement,
5. eligibleBot checks the daily cap,
6. eligibleBot checks available bonus pool balance,
7. eligibleBot calls the PublicSale daily bonus function,
8. PublicSaleGENc executes the bonus transfer.

This means the daily bonus mechanism is:

LP-purchase observed, bot-triggered and contract-executed.

The bot observes and triggers.

The contract executes.

10.4 PublicSale Phases

The PublicSale system uses phase-based participation logic.

The historical model includes:

- four phases,
- scaling minimum buy requirements,
- daily bonus amounts,
- daily wallet caps,
- rollover logic for unused bonus capacity.

The final public whitepaper should present exact phase values only after confirmation against the current deployed contract and current bot configuration.

PublicSale phases define different participation thresholds and daily bonus conditions.

10.5 DividendGENc Connection

PublicSaleGENc also acts as the execution layer for DividendGENc distributions.

DividendGENc is a technical distribution mechanism using:

- holder tracking,
- cycle snapshots,
- balance monitoring,
- initialHold logic,
- adjusted payout calculation,
- contract-executed GENc transfers.

The off-chain verifier calculates and submits the payout list.

The PublicSale contract executes the transfer.

DividendGENc is a cycle-based GENc distribution mechanism executed through PublicSaleGENc.

10.6 Airdrop Connection

PublicSaleGENc also supports list-based airdrop execution.

The airdrop system uses:

- holder tracking,
- prepared recipient lists,
- fixed or proportional distribution modes,
- batch execution through the PublicSale contract.

This means the airdrop system can use automated tracking while final execution may be script-driven. Tracking can be automated, while final airdrop execution may be CLI/script-triggered.

10.7 Contract-Executed, Bot-Triggered Model

The PublicSale system must be described with the same execution rule used across GENESIScoin:

Contracts execute. Bots trigger. Operators activate.

PublicSaleGENc holds the pools and performs transfers.

Bots and scripts:

- observe LP purchases,
- build holder lists,
- calculate cycle payouts,
- prepare airdrop batches,
- and submit transactions.

The contract defines what can be executed.

10.8 PublicSale Summary

PublicSaleGENc is the post-PreSale distribution module.

It connects PublicSale activation, daily bonus logic, DividendGENc distributions and airdrop execution into one contract-controlled layer.

It is not only a public sale contract.

It is the distribution execution layer for multiple post-PreSale mechanisms.

PublicSaleGENc holds and transfers post-PreSale distribution pools. Eligibility observation and list preparation may occur off-chain, but final transfer execution happens through the contract.

11. B100D Loyalty System

B100D is the loyalty-based holding mechanism connected to the GENESIScoin PreSale system.

It is designed to reward participants who enter the B100D whitelist, maintain their required holding condition and remain eligible through the active B100D period.

B100D is not a separate token.

It is a contract-connected loyalty mechanism using:

- PreSale state,
- whitelist registration,
- initialHold tracking,
- daily payout logic,
- balance verification,
- ejection logic,
- and final bonus distribution.

B100D is contract-executed and bot-supported.

The PreSale contract stores participant state and executes payout logic.

The B100D bot observes balances and triggers ejection if a participant falls below the required holding condition.

11.1 B100D Purpose

B100D exists to create a holding-based participation layer during the transition from PreSale into post-PreSale execution.

It is designed around four core principles:

1. participants must register,
2. participants must maintain their required holding level,
3. eligible participants receive daily GENc distributions,
4. eligible participants may receive a final bonus after the B100D period completes.

The mechanism is not based only on registration.

Registration is the entry point.

Holding behavior is the condition.

11.2 B100D Entry Flow

The B100D entry flow begins during the final PreSale stage.

The sequence is:

1. Stage 5 begins,
2. Oracle opens the B100D whitelist,
3. eligible participants register,
4. the contract records participant data,
5. PreSale later ends,
6. whitelist close occurs after the required window,
7. B100D starts after the required delay.

Stage 5 does not mean B100D has started.

Stage 5 means the B100D whitelist registration path is open.

B100D itself starts later, after whitelist closure and the required timing condition.

11.3 InitialHold

When a participant enters the B100D system, their holding condition is recorded as an initialHold reference.

This initialHold becomes the baseline used to determine whether the participant remains eligible.

If the participant's balance later drops below the required initialHold condition, the participant can be removed from active B100D eligibility.

This creates the core rule:

B100D rewards holding consistency, not only entry.

11.4 B100D Balance Enforcement

The B100D bot supports participant enforcement.

It observes registered participants and token transfer activity. If a tracked participant's balance falls below their recorded initialHold, the bot calls the contract ejection function.

The contract then updates the participant state.

This means the enforcement model has two layers:

Layer	Function
Contract	Stores participant status and executes ejection state changes
B100D Bot	Observes balances and triggers ejection transactions

The contract defines and executes the participant state. The bot observes balance changes and triggers enforcement when required.

The bot does not control the rules.

It triggers the contract-defined enforcement path.

11.5 Daily B100D Payout

B100D includes a daily GENc payout path for active eligible participants.

The Oracle is responsible for triggering the daily payout function when the required time condition is reached.

The contract executes the payout logic.

Daily B100D payouts are contract-executed and Oracle-triggered.

The historical public description refers to a daily payout amount of 15,000 GENc per participant and a final bonus proportional to initialHold. The final technical document should preserve the exact implemented day-count and payout logic from the deployed contract and automation layer.

11.6 Final B100D Bonus

After the B100D daily period reaches completion, the final bonus path becomes available.

The final bonus is triggered by a dedicated final bonus bot after the completion condition and configured delay.

The contract then processes the final bonus distribution.

The final B100D bonus is triggered after B100D completion and executed through the PreSale contract.

The final bonus is not a manual promise.

It is part of the B100D execution path.

11.7 B100D Summary

B100D is a holding-based loyalty mechanism integrated with the PreSale system.

It combines:

- whitelist registration,
- initialHold recording,
- daily payout execution,
- balance monitoring,
- ejection if holding falls below the required baseline,
- and final bonus distribution.

B100D rewards eligible participants who register, remain active and maintain the required holding condition through the B100D period.

12. DividendGENc System

DividendGENc is a cycle-based GENc token distribution mechanism.

It is not a legal dividend.

It is not equity.

It is not profit participation.

It is not yield.

It is not income rights.

DividendGENc is a technical system label used for GENc distributions from a dedicated pool through a tracker, verifier and contract execution path.

DividendGENc is a technical GENc distribution mechanism from a dedicated pool.

12.1 DividendGENc Purpose

DividendGENc exists to create a structured holder-based GENc distribution system.

The mechanism is designed around:

- holder tracking,
- minimum holding requirements,
- cycle snapshots,
- initialHold references,
- balance monitoring,
- adjusted payout calculation,
- and contract-executed transfers.

The system uses off-chain scripts to track and verify eligibility, while the PublicSale contract executes the final transfers.

12.2 Two Different Reward Layers

GENESIScoin has historically described two different reward layers:

Layer	Description
Rewards in BNB	Token-level reward mechanism connected to transaction-tax logic
DividendGENc	Separate GENc distribution mechanism from a dedicated GENc pool

These two mechanisms should not be merged.

Rewards in BNB are part of the token-level economic model.

DividendGENc is a separate cycle-based GENc distribution system using tracker and verifier scripts.

12.3 DividendTracker

DividendTracker maintains the holder set used by the DividendGENc system.

It observes relevant events and balance changes, including:

- PreSale purchase events,
- token transfer events,
- whitelist close / post-sale transition signals,
- balance updates,
- minimum holding filters,
- excluded addresses.

The output of this tracking layer is a holder file used by the verifier.

DividendTracker maintains the eligible holder dataset used by the DividendGENc verification layer.

12.4 DividendVerifier

DividendVerifier creates and manages DividendGENc cycles.

Its role is to:

1. create a cycle snapshot,
2. read tracked holders,
3. verify balances,
4. record initialHold values,
5. monitor balance changes during the cycle,
6. remove users who fall below the required condition,
7. apply adjusted initialHold logic where applicable,
8. calculate payout amounts,
9. submit the payout batch to PublicSaleGENc.

The verifier does not hold the dividend pool.

The PublicSale contract holds the pool and executes the transfer.

12.5 InitialHold and Balance Maintenance

DividendGENc uses an initialHold model.

At cycle entry, the user's qualifying balance becomes the reference for that cycle.

To remain eligible, the user must maintain the required balance condition through the active cycle.

If the user falls below the required threshold or below their cycle reference, they can be removed from that cycle.

DividendGENc eligibility is based on cycle entry, balance reference and continued holding during the cycle.

12.6 Adjusted Balance Rule

DividendGENc includes an adjusted balance concept.

If a holder significantly increases their balance during an active cycle, the system can update the initialHold reference to reflect the increased holding level.

This means future payout calculations can use the updated balance reference rather than the original lower reference.

DividendGENc can update a holder's cycle reference when the holder materially increases their balance under the defined rule.

This must be described as technical mechanics, not financial return language.

12.7 Payout Execution

DividendVerifier calculates the payout list.

PublicSaleGENc executes the payout transfer.

The model is:

Layer	Function
DividendTracker	Maintains holder dataset
DividendVerifier	Creates cycles and calculates payout lists
PublicSaleGENc	Holds the pool and executes GENc transfers

DividendGENc payouts are verifier-calculated and contract-executed.

12.8 DividendGENc Summary

DividendGENc is a technical GENc distribution system.

It combines off-chain eligibility tracking with on-chain payout execution.

DividendGENc uses cycle snapshots, holder verification and PublicSale contract execution to distribute GENc from a dedicated pool to eligible holders.

13. Airdrop System

The GENESIScoin airdrop system is a tracked-holder and list-based distribution mechanism.

It combines automated holder tracking with script-based execution through the PublicSale contract.
The system is not purely automatic.
Tracking can be automated.
Final execution may be CLI/script-triggered.
The airdrop system tracks eligible holders and executes prepared distributions through PublicSaleGENc.

13.1 Airdrop Purpose

The airdrop system exists to support structured GENc distributions for:

- holder recognition,
- campaign participation,
- ecosystem activity,
- milestone-based distribution,
- selected distribution lists,
- proportional or fixed allocations.

It is not a random emission mechanism.

It is a controlled distribution path connected to tracked holder data and contract-executed transfer functions.

13.2 Airdrop Tracker

The airdrop tracker observes holder activity and maintains a tracked holder dataset.

It can track users from:

- PreSale purchase activity,
- token transfers,
- post-sale LP purchase activity,
- balance checks,
- minimum holding rules,
- excluded address lists.

The output is a tracked holder file used by the airdrop execution script.

The airdrop tracker maintains the holder dataset used for later distribution preparation.

13.3 Airdrop Console

AirdropConsole is a manual CLI execution tool.

It can prepare distribution lists based on tracked holder data and execute batch airdrops through the PublicSale contract.

It supports distribution modes such as:

- fixed amount per eligible wallet,
- proportional distribution based on holder balances.

The script builds recipient and amount arrays, then calls the PublicSale contract's batch airdrop function.

AirdropConsole is a script-driven execution tool, not a permanent daemon.

13.4 Airdrop Execution Model

The airdrop model is:

1. holders are tracked,
2. eligible list is prepared,
3. distribution mode is selected,
4. recipients and amounts are calculated,
5. airdrop batch is submitted,
6. PublicSaleGENc executes the transfer.

This separates observation from execution.

Layer	Function
tracker.js	Tracks eligible holders
trackedHolders.json	Stores holder dataset
airdropConsole.js	Builds and submits distribution batch
PublicSaleGENc	Executes the airdrop transfer

13.5 Airdrop Summary

The GENESIScoin airdrop system is a structured distribution mechanism.

It is designed to combine holder tracking, prepared lists and contract-executed batch transfers.

GENESIScoin airdrops are list-based distributions executed through PublicSaleGENc using tracked holder data and script-prepared batches.

14. Asset Manager v1

Asset Manager v1 is the fixed-schedule treasury release layer of GENESIScoin.

It controls long-term ecosystem allocations through predefined schedules.

The purpose of Asset Manager v1 is to reduce discretionary treasury movement by placing defined allocations inside a contract-controlled release structure.

Asset Manager v1 releases are contract-defined and bot-triggered.

14.1 Asset Manager v1 Purpose

Asset Manager v1 controls the fixed-schedule allocations for ID1-ID6:

ID	Pool
ID1	Team GENc
ID2	Marketing GENc
ID3	Development GENc
ID4	Event / Expo
ID5	Future Project
ID6	SBA Core / Reserve

These allocations represent the long-term treasury and ecosystem reserve structure of GENESIScoin.

The total controlled allocation for ID1-ID6 is:

20,200,000,000 GENc

14.2 Full-Funding Principle

The Asset Manager v1 model is based on full funding before activation.

Each ID has:

- a defined cap,
- a funded amount,
- a released amount,
- a sealed state,
- a release schedule,
- and a destination/origin wallet.

The intended safety rule is:

pools are fully funded and sealed before WasteStart.

This prevents partial release logic and keeps the contract release model deterministic.

14.3 WasteStart Anchor

WasteStart is the fixed timestamp anchor for the Asset Manager v1 release schedules.

The system flow is:

1. PreSale whitelist closes,
2. the bot detects the WhitelistClosed timestamp,
3. the bot calls setWasteStart,
4. WasteStart is recorded,
5. all ID1-ID6 release schedules anchor to that timestamp,
6. bot checks each ID periodically,
7. due tranches are released by contract call.

WasteStart is set from the PreSale WhitelistClosed timestamp and becomes the fixed anchor for Asset Manager v1 release schedules.

14.4 Release Execution

Releases are performed by an authorized bot.

The bot checks whether a tranche is due.

If the tranche is due, the bot calls the release function.

The contract then:

- verifies WasteStart,
- verifies pool state,
- verifies due timing,
- transfers the defined amount,
- updates released state,
- advances the nextIndex.

The bot does not choose an arbitrary amount.

The bot does not choose a new recipient.

The contract defines the amount and recipient.

14.5 ID1-ID6 Release Schedules

ID1 — Team GENc

Step	Amount	Timing
TGE	500,000,000 GENc	At WasteStart
Cliff	—	100 days
Scheduled tranches	300,000,000 GENc × 15	Every 30 days after cliff

Total: 5,000,000,000 GENc

ID2 — Marketing GENc

Step	Amount	Timing
TGE	500,000,000 GENc	At WasteStart
Cliff	—	75 days
Scheduled tranches	300,000,000 GENc × 15	Every 30 days after cliff

Total: 5,000,000,000 GENc

ID3 — Development GENc

ID3 has no TGE release.

Step	Amount	Timing
Release 1	200,000,000 GENc	90 days after WasteStart
Release 2	200,000,000 GENc	50 days after previous release
Release 3	100,000,000 GENc	50 days after previous release
Release 4	100,000,000 GENc	50 days after previous release
Release 5	100,000,000 GENc	50 days after previous release
Release 6	100,000,000 GENc	50 days after previous release
Release 7	200,000,000 GENc	60 days after previous release

Total: 1,000,000,000 GENc

ID4 — Event / Expo

ID4 has no TGE release.

Step	Amount	Timing
Release 1	1,000,000,000 GENc	365 days after WasteStart
Release 2	1,000,000,000 GENc	365 days after previous release
Release 3	1,000,000,000 GENc	365 days after previous release

Total: 3,000,000,000 GENc

ID5 — Future Project

ID5 has no TGE release.

Step	Amount	Timing
Release 1	3,100,000,000 GENc	365 days after WasteStart

Total: 3,100,000,000 GENc

ID6 — SBA Core / Reserve

Step	Amount	Timing
TGE	600,000,000 GENc	At WasteStart
Release 1	500,000,000 GENc	50 days after previous release
Release 2	500,000,000 GENc	50 days after previous release
Release 3	500,000,000 GENc	50 days after previous release
Release 4	500,000,000 GENc	50 days after previous release
Release 5	500,000,000 GENc	50 days after previous release

Total: 3,100,000,000 GENc

14.6 Origin-Based Release Model

Asset Manager v1 uses a defined recipient model.

Each ID has one destination/origin wallet.

Releases always go to that predefined address.

This means the movement path is:

original system allocation → Asset Manager v1 → predefined origin wallet by schedule

The release path is not an open withdrawal system.

14.7 Bot Loop

The Asset Manager v1 bot operates as a release trigger.

The intended loop model is:

1. check if WasteStart is set,
2. if not set, listen for WhitelistClosed,
3. once WasteStart exists, loop through ID1-ID6,
4. check whether the next tranche is due,
5. call release if due,
6. skip if not due,
7. repeat periodically.

The official plan describes a six-hour operational loop for release checks.

The bot checks the schedule periodically and triggers due releases. The contract decides whether a release is valid.

14.8 Asset Manager v1 Summary

Asset Manager v1 is the fixed treasury release layer of GENESIScoin.

It controls ID1-ID6 allocations through deterministic schedules anchored to WasteStart.

Asset Manager v1 controls fixed-schedule treasury allocations. Pools are funded, sealed and released to predefined origin wallets according to contract-defined timing, with bot-triggered execution.

This module separates long-term treasury release from daily operational activity and public distribution systems.

15. Asset Manager v2

Asset Manager v2 is the conditional operational execution controller of the GENESIScoin system.

It is separate from Asset Manager v1.

Asset Manager v1 controls fixed treasury release schedules.

Asset Manager v2 controls dynamic operational flows based on stage outcomes, approved operations, cliffs and predefined routing paths.

This makes AMv2 a higher-level execution module than a passive reserve wallet.

It does not only hold GENc.

It defines how selected GENc pools can move, when they can move and where they are routed.

15.1 AMv2 Purpose

Asset Manager v2 exists to separate dynamic ecosystem execution from fixed treasury vesting.

It controls operational GENc pools used for:

- stage-based airdrop execution,
- milestone-based ecosystem distribution,
- partner-related operational releases,
- contest and promotional distributions.

AMv2 should not be described as:

- Asset Manager v1,
- RM01,
- LP Main,
- LP Backup,
- PublicSaleGENc,
- a general treasury wallet,
- or a discretionary token wallet.

Asset Manager v2 is a contract-controlled operational routing module.

15.2 AMv2 Pool Structure

Asset Manager v2 defines four internal pool IDs.

Pool ID	Pool	Execution model	Primary destination
1	Stage Airdrop Pool	GENERAL-triggered release or fail path	Airdrop wallet, or Burn + Dividend wallets on fail
2	Milestone Pool	Owner proposal + 2-of-4 multisig approval + execution	Airdrop wallet
3	Partners Pool	Owner proposal + 2-of-4 multisig approval + arm + cliff + execution	Partners wallet
4	Contest / Promotional Pool	Owner proposal + 2-of-4 multisig approval + arm + cliff + execution	Airdrop wallet

The Stage Airdrop Pool and Milestone Pool have hardcoded caps:

Pool	Cap
Stage Airdrop Pool	1,500,000,000 GENc
Milestone Pool	1,000,000,000 GENc

The Partners Pool and Contest / Promotional Pool are funded through contract funding actions but do not use the same hardcoded cap model as the Stage and Milestone pools.

For publication, AMv2 pool balances should always be reconciled against current on-chain state.

The current visible AMv2 contract balance is treated as a custody snapshot, not as a replacement for the internal pool accounting model.

15.3 AMv2 Accounting Model

AMv2 tracks pool state internally.

Each pool can have:

- funded amount,
- released amount,
- current pool balance.

This means AMv2 should be read through both views:

View	Meaning
Contract token balance	Total visible GENc held by AMv2
Internal pool balance	Amount available inside each AMv2 pool
Released amount	Amount already executed from a pool
Stage / milestone flags	Records preventing duplicate release

A visible AMv2 token balance alone does not explain the full internal state.

The internal pool accounting determines what can still be released.

15.4 Stage Airdrop Pool

The Stage Airdrop Pool is the AMv2 pool connected to stage outcomes.

It is executed only by the GENERAL role.

This pool has a total cap of:

1,500,000,000 GENc

The stage amounts are fixed:

Stage	Amount
Stage 1	100,000,000 GENc
Stage 2	200,000,000 GENc
Stage 3	300,000,000 GENc
Stage 4	400,000,000 GENc
Stage 5	500,000,000 GENc

Each stage can be processed only once.

AMv2 records whether a stage has already been released or failed.

This prevents duplicate execution for the same stage.

15.5 Stage PASS Path

When a stage passes, the GENERAL role triggers the stage release path.

The result is:

Outcome	Routing
Stage PASS	100% of the stage amount is sent to the Airdrop wallet

This means a successful stage routes the defined stage allocation to the Airdrop wallet.

The contract:

- checks that the stage ID is valid,
- checks that the stage was not already processed,
- checks that the Stage Airdrop Pool has enough balance,
- marks the stage as processed,
- decreases the Stage Airdrop Pool balance,
- increases the released amount,
- transfers the stage amount to the Airdrop wallet,
- emits execution events.

This is not a manual distribution.

It is a contract-defined routing path triggered by the authorized execution role.

15.6 Stage FAIL Path

AMv2 also defines a fail path for eligible stages.

The fail path applies only to:

- Stage 2,
- Stage 3,
- Stage 4.

Stage 1 and Stage 5 do not use the same fail path.

When an eligible stage fails, the GENERAL role triggers the fail execution path.

The result is:

Outcome	Routing
Stage FAIL	50% to Burn wallet and 50% to Dividend wallet

This means failed stage allocation is not simply left idle.

It is redirected.

The contract:

- checks that the stage is within the fail-eligible range,
- checks that the stage was not already processed,
- checks that the Stage Airdrop Pool has enough balance,
- marks the stage as processed,

- decreases the Stage Airdrop Pool balance,
- increases the released amount,
- splits the stage amount,
- transfers part to the Burn wallet,
- transfers part to the Dividend wallet,
- emits fail-split and release events.

The fail path creates a different outcome from a normal release.

A successful stage routes GENc to the Airdrop wallet.

A failed eligible stage routes GENc into burn and DividendGENc-related distribution paths.

This is the core reason AMv2 should be described as a conditional routing layer.

15.7 Stage Routing Summary

AMv2 stage execution can be summarized as:

Stage result	Destination
PASS	Airdrop wallet
FAIL, Stage 2-4 only	Burn wallet + Dividend wallet
Already processed	Cannot execute again
Insufficient pool balance	Execution rejected
Invalid stage path	Execution rejected

This makes AMv2 reactive to stage outcome.

It does not treat all stage allocations as simple airdrop reserves.

15.8 Milestone Pool

The Milestone Pool is used for milestone-based ecosystem distribution.

It has a total cap of:

1,000,000,000 GENc

Milestone releases do not use the GENERAL role.

They use a proposal-and-approval execution model.

The valid milestone IDs and amounts are:

Milestone ID	Amount
100	20,000,000 GENc
250	60,000,000 GENc
500	140,000,000 GENc
1000	300,000,000 GENc
1500	480,000,000 GENc

Total milestone allocation:

1,000,000,000 GENc

Each milestone can be released only once.

The execution flow is:

1. owner proposes the milestone release,
2. the milestone ID is checked,
3. the milestone must not already be released,
4. the Milestone Pool must have enough balance,
5. two distinct multisig approvals are required,
6. the approved operation becomes executable,
7. AMv2 transfers the milestone amount to the Airdrop wallet.

This creates a controlled milestone distribution path.

It is not an open wallet withdrawal.

15.9 Partners Pool

The Partners Pool is used for controlled partner-related execution paths.

It does not release in a single direct step.

It uses an arm-and-execute model with a cliff.

The destination for executed Partners Pool releases is the Partners wallet:

0x748D87424ee8551AAD6B7150A22438996a039F91

The Partners execution flow is:

1. owner proposes a Partners arm operation,
2. the proposed amount must be greater than zero,
3. the Partners Pool must have enough balance,
4. two distinct multisig approvals are required,
5. the approved arm operation is executed,
6. the selected amount is removed from available pool balance,
7. the amount becomes armed,
8. a 240-hour cliff starts,
9. after the cliff, owner proposes Partners execution,
10. two distinct multisig approvals are required again,
11. the approved execution transfers the armed amount to the Partners wallet.

The Partners Pool also supports cancellation.

If an armed Partners operation is cancelled:

- the armed state is cleared,
- the amount returns to the Partners Pool balance,
- no transfer to the Partners wallet happens.

This means the Partners Pool has three distinct states:

State	Meaning
Available	Amount remains inside Partners Pool balance
Armed	Amount is reserved and waiting for cliff completion
Executed	Amount is transferred to Partners wallet
Cancelled	Armed amount returns to Partners Pool balance

This creates a delayed and approval-gated partner release path.

15.10 Contest / Promotional Pool

The Contest / Promotional Pool supports controlled campaign, contest and promotional distributions.

It uses a similar arm-and-execute model to the Partners Pool, but with a shorter cliff.

The destination for executed Contest / Promotional releases is the Airdrop wallet.

The Contest execution flow is:

1. owner proposes a Contest arm operation,
2. the proposed amount must be greater than zero,
3. the Contest Pool must have enough balance,
4. two distinct multisig approvals are required,
5. the approved arm operation is executed,
6. the selected amount is removed from available pool balance,
7. the amount becomes armed,
8. a 72-hour cliff starts,
9. after the cliff, owner proposes Contest execution,
10. two distinct multisig approvals are required again,
11. the approved execution transfers the armed amount to the Airdrop wallet.

The Contest Pool also supports cancellation.

If an armed Contest operation is cancelled:

- the armed state is cleared,
- the amount returns to the Contest Pool balance,
- no transfer to the Airdrop wallet happens.

This separates campaign-related execution from direct manual distribution.

15.11 Multisig Approval Model

AMv2 uses a 2-of-4 multisig approval model for proposal-based operations.

This approval model applies to:

- milestone release,
- Partners arm,
- Partners execute,

- Partners cancel,
- Contest arm,
- Contest execute,
- Contest cancel.

The proposal path is owner-initiated.

The approval path requires two distinct multisig signers.

Once an operation has two approvals, execution can occur through the contract execution path.

The execution path checks that:

- the operation exists,
- the operation has not already been executed,
- at least two approvals exist,
- the operation type is valid,
- pool balance and state conditions are satisfied.

This creates separation between:

Step	Role
Proposal	Owner creates an operation
Approval	Multisig signers approve
Execution	Contract executes the approved operation
Event record	Contract emits proof of action

AMv2 should therefore not be described as fully discretionary owner control.

The owner proposes, but selected operations require multisig approval before execution.

15.12 AMv2 Roles and Destinations

AMv2 uses defined roles and destination wallets.

Component	Meaning
Owner	Proposes and funds selected operations
GENERAL	Executes Stage Pool release or fail paths
Multisig committee	Approves proposal-based operations
Airdrop wallet	Receives stage-pass, milestone and contest execution outputs
Partners wallet	Receives executed partner releases
Burn wallet	Receives burn-side stage-fail output
Dividend wallet	Receives dividend-side stage-fail output

The important distinction is that each pool has a different destination model.

AMv2 is not one general-purpose transfer wallet.

15.13 Pool Sealing Interpretation

AMv2 contains a pool-sealed view for the capped pools.

The Stage Airdrop Pool is considered sealed when it is funded to its full 1.5B GENc cap.

The Milestone Pool is considered sealed when it is funded to its full 1B GENc cap.

Partners and Contest pools do not return the same sealed state under this model.

This means:

Pool	Seal interpretation
Stage Airdrop Pool	Sealed when funded to full cap
Milestone Pool	Sealed when funded to full cap
Partners Pool	Not sealed by the same cap model
Contest / Promotional Pool	Not sealed by the same cap model

The final whitepaper should avoid saying that all AMv2 pools are sealed.

Only the capped pools use that seal interpretation.

15.14 AMv2 and RM01 Separation

AMv2 and RM01 must remain separate in documentation.

AMv2 contains a Partners Pool that can route GENc to the Partners wallet through an approval-gated and cliff-based execution path.

RM01 is a separate deployed contract with its own:

- pool,
- funding state,
- seal state,
- TGE reference,
- recipient path,
- RM_BOT-triggered schedule.

AMv2 does not execute RM01 schedules.

RM01 does not define AMv2 stage, milestone, partners or contest logic.

Allocation-wise, RM01 can belong to the broader Partners allocation model.

Contract-wise, RM01 is separate from AMv2.

This distinction prevents double-counting and prevents incorrect system interpretation.

15.15 AMv2 Execution Summary

Asset Manager v2 has four main execution paths:

Area	Execution path	Output
Stage Airdrop Pool	GENERAL-triggered pass/fail execution	Airdrop wallet, or Burn + Dividend wallets
Milestone Pool	Owner proposal + 2-of-4 multisig approval + execution	Airdrop wallet
Partners Pool	Owner proposal + 2-of-4 multisig approval + arm + 240h cliff + execution	Partners wallet
Contest / Promotional Pool	Owner proposal + 2-of-4 multisig approval + arm + 72h cliff + execution	Airdrop wallet

This is the core AMv2 model.

It is a conditional execution controller.

It combines:

- fixed stage amounts,
- pass/fail routing,
- milestone-based release amounts,
- proposal-based execution,
- multisig approvals,
- cliff-based armed releases,
- defined destination wallets,
- and on-chain event records.

Asset Manager v2 controls dynamic GENc operational flows through contract-defined routing logic. It is not a passive reserve wallet and it is not RM01. It is the conditional operational execution layer of the GENESIScoin system.

16. RM01

RM01 is a separate strategic release module inside the GENESIScoin system.

It is deployed.

It is verified.

It was funded with 50,000,000 GENc.

Its pool was sealed.

RM01 is not pending.

RM01 must be described as part of the current deployed system.

16.1 RM01 Purpose

RM01 exists to control the first 50,000,000 GENc strategic release schedule from the RM01-RM10 reserve.

The RM01-RM10 reserve is a 500,000,000 GENc sub-allocation inside the 3,000,000,000 GENc Partners allocation.

RM01 is therefore not an additional allocation outside the original supply model.

It is the first deployed module in a planned ten-module RM sequence:

RM Module	Allocation
RM01	50,000,000 GENc

RM Module	Allocation
RM02	50,000,000 GENc
RM03	50,000,000 GENc
RM04	50,000,000 GENc
RM05	50,000,000 GENc
RM06	50,000,000 GENc
RM07	50,000,000 GENc
RM08	50,000,000 GENc
RM09	50,000,000 GENc
RM10	50,000,000 GENc

Total RM01–RM10 reserve: 500,000,000 GENc.

It is not a general treasury wallet.

It is not a liquidity wallet.

It is not the Asset Manager v2 Partners Pool.

It is a dedicated release module funded from the reserved RM layer inside the Partners allocation, with:

- fixed pool size,
- sealed funding,
- hardcoded recipient path,
- TGE trigger,
- scheduled release logic,
- RM_BOT-triggered execution.

16.2 RM01 Identity

Field	Value
Module	RM01
Address	0x40A830b22E4C39D16B8b17e8beD4ED0f7C29E547
Pool	50,000,000 GENc
Allocation source	RM01–RM10 reserve inside the 3B GENc Partners allocation
Funding source	Partners wallet / Partners allocation
Status	Funded and sealed
Executor	RM_BOT
Release model	TGE + scheduled releases

16.3 RM01 Funding and Seal

RM01 is funded from the Partners wallet / Partners allocation.

The 50,000,000 GENc RM01 pool belongs to the 500,000,000 GENc RM01–RM10 reserve inside the 3,000,000,000 GENc Partners allocation.

After the full 50,000,000 GENc RM01 allocation is funded, the pool is sealed.

The sealed state means that RM01 is not treated as an open-ended funding container.

It has a defined cap and a defined release structure.

RM01 is a funded and sealed 50M GENc strategic release module.

16.4 RM01 TGE

RM01 uses a TGE reference connected to the PreSale / whitelist close state.

The execution model is:

1. RM01 is funded,
2. RM01 is sealed,
3. PreSale whitelist close timestamp exists,
4. RM_BOT calls SET_TGE,
5. RM01 records the TGE timestamp,
6. the release schedule begins.

RM01 TGE is RM_BOT-triggered from the PreSale whitelist-close reference.

16.5 RM01 Release Schedule

RM01 controls a total of 50,000,000 GENc.

The schedule is:

Step	Amount
TGE	10,000,000 GENc
Release 1	5,000,000 GENc
Release 2	5,000,000 GENc
Release 3	5,000,000 GENc
Release 4	5,000,000 GENc
Release 5	5,000,000 GENc
Release 6	5,000,000 GENc
Release 7	5,000,000 GENc
Release 8	5,000,000 GENc

Total: **50,000,000 GENc**

16.6 RM01 Execution

RM01 releases are triggered by RM_BOT.

The bot reads RM01 state, checks whether a release is due and calls the release function when the contract schedule allows it.

The contract defines:

- whether TGE is set,
- the next release index,
- the next due timestamp,
- the next amount,
- total released amount,
- sealed pool state.

RM01 releases are contract-defined and RM_BOT-triggered.

The bot does not create a new schedule.

The bot triggers the schedule that the contract defines.

16.7 RM01 Separation Rule

RM01 must stay separate in documentation.

Do not describe RM01 as:

- LP Main,
- LP Backup,
- LP Manager,
- Asset Manager v1,
- Asset Manager v2,
- PublicSale,
- a general partner wallet.

RM01 is the first standalone 50M GENc strategic release module from the 500M GENc RM01-RM10 reserve inside the 3B GENc Partners allocation.

This means RM01 is separate at the contract level, but not separate from the Partners allocation at the tokenomics level.

16.8 RM01 Summary

RM01 adds a dedicated strategic release layer to the GENESIScoin system.

It is the first deployed module from the RM01-RM10 reserve.

The RM01-RM10 reserve equals 500,000,000 GENc and sits inside the broader 3,000,000,000 GENc Partners allocation.

RM01 controls the first 50,000,000 GENc from that reserve through a sealed contract module, TGE reference and scheduled release execution.

RM01 is the first deployed, verified, funded and sealed strategic release module from the RM01-RM10 reserve inside the Partners allocation.

This prevents double-counting and keeps the allocation model consistent.

17. Liquidity Layer

The GENESIScoin liquidity layer is based on defined liquidity allocations and a staged activation path.

The original allocation model includes two major liquidity allocations:

- LP Main,
- LP Backup.

Together, they represent 20,000,000,000 GENc.

These allocations are central to the future liquidity and trading layer of the system.

17.1 LP Main

LP Main is the primary liquidity allocation.

Field	Value
Allocation	LP Main
Amount	10,000,000,000 GENc
Purpose	Primary liquidity deployment allocation
Current classification	Liquidity allocation wallet

LP Main is intended to support the main liquidity path for GENESIScoin.

This may include initial liquidity deployment, trading-pair support and primary DEX liquidity structure, depending on the final activation path and verifiable transactions.

17.2 LP Backup

LP Backup is the secondary liquidity allocation.

Field	Value
Allocation	LP Backup
Amount	10,000,000,000 GENc
Purpose	Backup / secondary liquidity reserve
Current classification	Liquidity allocation wallet

LP Backup exists to support future liquidity expansion, contingency liquidity needs or additional liquidity deployment paths.

17.3 Liquidity Allocation vs Active LP

A liquidity allocation is not the same as an active liquidity pair.

This distinction is critical.

Term	Meaning
LP Main	Defined liquidity allocation wallet
LP Backup	Defined backup liquidity allocation wallet
Active LP pair	Live DEX pair used for trading
LP Manager	Future liquidity governance module
Locked liquidity	A separate custody state that must be proven on-chain

The final whitepaper must not treat these as the same thing.

LP Main and LP Backup are defined liquidity allocations. Active liquidity deployment must be verified through actual liquidity transactions and pair state.

17.4 Stage 1 Liquidity / Trading Threshold

Stage 1 reaching 50% sold is the first major threshold connected to the liquidity/trading activation path.

Stage 1 50% sold creates the structural condition for the liquidity/trading activation path.

17.5 LP Manager

The LP Manager is the planned liquidity-governance smart-contract layer.

At the current stage, it must be described as future infrastructure unless deployed and funded on-chain.

The LP Manager is planned as a future liquidity-governance smart-contract module. It should not be described as controlling LP Main or LP Backup until it is deployed and custody transfer is verifiable on-chain.

17.6 Liquidity Layer Summary

The liquidity layer contains:

- LP Main,
- LP Backup,
- staged liquidity/trading activation conditions,
- future LP Manager / liquidity-governance contract infrastructure,
- and later active liquidity deployment records.

GENESIScoin separates liquidity allocation from liquidity execution. LP Main and LP Backup define the allocation layer; active liquidity deployment and LP Manager custody must be verified when executed on-chain.

18. Execution Model

GENESIScoin uses a hybrid execution model.

The system combines:

- deterministic smart contracts,
- bot-triggered transactions,
- operator-managed activation,
- public blockchain state,
- and phase-dependent execution.

Contracts execute. Bots trigger. Operators activate.

This phrase should be treated as a core documentation rule.

18.1 Contract Layer

The contract layer defines:

- who can call specific functions,
- what state is required,
- when a transfer is valid,
- how pool balances are tracked,
- how release schedules work,
- what happens when a function executes,
- and what events are emitted.

Contracts are the execution authority.

They either accept or reject a transaction according to their logic.

18.2 Bot Layer

The bot layer supports timing and triggering.

Bots and scripts:

- observe events,
- read contract state,
- monitor timestamps,
- check balances,
- prepare transactions,
- submit calls,
- and maintain local operational files where required.

Bots do not replace contract rules.

They trigger functions that contracts allow.

Bots are transaction triggers for contract-defined logic.

18.3 Operator Layer

The operator layer is responsible for infrastructure and activation.

Operators:

- deploy contracts,
- fund pools,

- configure scripts,
- start bots before their activation windows,
- monitor logs,
- maintain RPC access,
- ensure gas availability,
- and manage operational continuity.

This means the system is not described as fully automatic without dependency.

It is described as:

contract-enforced, bot-triggered and operator-activated.

18.4 Phase-Dependent Automation

GENESIScoin automation is phase-dependent.

Not every bot must run permanently.

Some bots are broadly active during major system phases.

Some bots are activated one stage before they are needed.

Some scripts are manual CLI tools.

Some processes are cycle-based.

Example:

If a bot must act in Stage 5, it may be started in Stage 4 so that it is already listening when Stage 5 begins.

This creates three operational states:

1. pre-activation,
2. listening,
3. trigger execution.

Bot runtime must be interpreted against the current system phase.

18.5 Runtime Interpretation

A stopped bot does not automatically mean that a system path is invalid.

It may mean:

- the bot is outside its activation window,
- the relevant system phase has not started,
- the bot is manual/CLI by design,
- or the process is intentionally inactive until needed.

However, a bot not running during its required activation window can create execution delay.

A bot being inactive outside its activation window is not evidence of system failure. A bot being inactive inside its required execution window is an operational risk.

18.6 Execution Summary

GENESIScoin should be understood as a layered execution system:

Layer	Role
Smart contracts	Define and execute rules
Bots / scripts	Trigger contract functions and maintain observation logic
Operators	Activate and maintain runtime infrastructure
Public chain	Provides verifiable state and transaction history

This model is realistic and transparent.

It avoids false claims of full autonomy while still showing that execution is governed by deployed contract logic.

19. Risks and Limitations

This whitepaper describes the GENESIScoin system architecture and current execution model.

It does not remove risk.

GENESIScoin is a crypto-asset system using smart contracts, automation scripts, public-chain infrastructure and staged execution. Participation in any crypto-asset system involves risk.

This section is a public system limitation summary, not a legal document.

19.1 Smart Contract Risk

GENESIScoin relies on deployed smart contracts.

Smart contracts can contain bugs, unintended behavior, integration issues or vulnerabilities.

Verification of source code does not equal an audit and does not guarantee safety.

Verified source improves transparency. It does not eliminate technical risk.

19.2 Automation Risk

Some GENESIScoin execution paths require bots or scripts to submit transactions at the correct time.

If a bot is offline, misconfigured, unfunded for gas, using an incorrect RPC endpoint or started too late, execution may be delayed.

This affects timing, not necessarily the existence of the contract function.

Contract logic can exist even when the trigger process is not active.

19.3 Oracle and External Data Risk

The Oracle layer supports price updates, stage transitions, whitelist timing, B100D timing, BNB distribution and PublicSale activation.

If Oracle execution fails, is delayed or uses incorrect configuration, system transitions may be delayed or affected.

Oracle-triggered execution introduces operational dependency.

19.4 Tracking and File-State Risk

Some systems use off-chain tracking files and scripts.

Examples include:

- airdrop holder tracking,
- dividend holder tracking,
- cycle metadata,
- payout lists,
- bonus daily state,
- release bot state.

If local files are corrupted, stale or inconsistent, eligibility or execution data may require correction or resynchronization.

Off-chain tracking supports execution but must be maintained carefully.

19.5 Liquidity Risk

Liquidity is not guaranteed.

LP Main and LP Backup are defined liquidity allocations, but active market liquidity depends on actual deployment, pair creation, liquidity provisioning and market conditions.

A liquidity allocation is not the same as guaranteed tradability.

Liquidity must be verified through actual on-chain liquidity deployment.

19.6 LP Manager Pending Status

The LP Manager is not described as active unless deployed and funded.

Until that happens, it remains future infrastructure.

Planned infrastructure is not current custody.

19.7 Market and Participation Risk

GENc may experience volatility, low liquidity, no liquidity, low demand or complete loss of market value.

No system architecture guarantees market performance.

Verifiable execution does not guarantee price, liquidity, adoption or outcome.

19.8 No Outcome Guarantee

GENESIScoin documentation should not be interpreted as a guarantee of:

- profit,

- yield,
- trading,
- liquidity,
- exchange listing,
- price appreciation,
- reward continuity,
- or ecosystem success.

The project standard is Proof Before Promise, not promise before execution.

20. Closing Statement

GENESIScoin is built as a modular on-chain system on BNB Smart Chain.

Its architecture separates the ecosystem into distinct execution layers:

- the GENESIScoin token as the asset layer,
- PreSaleGENc as the staged initial distribution layer,
- PublicSaleGENc as the post-PreSale distribution layer,
- Asset Manager v1 as the fixed treasury release layer,
- Asset Manager v2 as the conditional operational execution controller,
- RM01 as a separate strategic release module,
- LP Main and LP Backup as defined liquidity allocations,
- and a future LP Manager as planned liquidity-governance smart-contract infrastructure.

The system is not presented as trustless by slogan.

It is presented as verifiable by structure.

The standard remains:

Proof Before Promise.

And the behavior remains:

Don't Assume. Verify.

GENESIScoin exists to show execution before narrative, deployed infrastructure before expansion claims, and system logic before promises.

The record should be checked.

The contracts should be read.

The flows should be verified.

That is the GENESIScoin standard.