IBITIcoin (IBITI) – Whitepaper

Version 1.0 – 2025
Network: BNB Smart Chain (BEP-20, chainId 56)

This document describes the IBITIcoin protocol as it is actually implemented on-chain: a modular DeFi system deployed on BNB Smart Chain, with a capped base token, on-chain treasury, dynamic fee engine, staking, NFT utilities and DAO governance. It is not a marketing brochure; the goal is to reflect the real architecture and operational model.

Contract source code is publicly available in the IBITIcoin GitHub repository. Where this document and the code differ, the code and on-chain configuration are the source of truth.

Nothing in this document constitutes financial, legal or tax advice. IBITIcoin is an experimental DeFi project. Participants should do their own research and never allocate funds they cannot afford to lose.

1. Vision & Motivation

Most small-cap tokens live in a split reality. On-chain, there is a simple ERC-20 contract. Off-chain, there are spreadsheets, promises, screenshots and private deals. Once the cycle turns, nothing of that off-chain structure is enforceable.

IBITIcoin was designed from the beginning as a "contract-first" project: if something affects fees, rewards, liquidity or governance, it should be encoded in transparent smart contracts, not only in social media posts. The system is built out of independent modules instead of a single monolithic contract.

The long-term vision is to turn IBITIcoin into a base asset for a small but serious ecosystem: tools for trading and holding, staking and NFT utility, DAO-managed parameters and an on-chain treasury that can be audited by anyone at any time.

2. Token definition & supply

Core parameters

• Name: IBITIcoin
• Symbol: IBITI
• Standard: BEP-20 (compatible with ERC-20)
• Network: BNB Smart Chain (BSC, chainId 56)
• Decimals: 8
• Maximum supply: 100,000,000 IBITI (hard cap in the token contract)
• Token contract (BscScan): 0x47F2FFCb164b2EeCCfb7eC436Dfb3637a457B9bb

2.1 Supply model

The IBITI token contract mints the full maximum supply at deployment. From there, supply is distributed into distinct buckets managed by contracts and wallets with different roles. There is no further mint function and no implicit inflation beyond the initial cap.

The high-level intent is to keep a significant portion of total supply in an on-chain treasury and vesting contracts, while circulating supply grows gradually through liquidity provision, staking rewards, ecosystem incentives and DAO-decided programs.

2.2 High-level allocation (conceptual)

Exact percentages are configurable and may evolve over time. On-chain contracts and treasury transactions always provide the final, verifiable state. Conceptually, supply is grouped as:
• Treasury and protocol reserves – the central on-chain pool used for liquidity, rewards and ecosystem funding.
• Liquidity and market making – allocations dedicated to seeding and reinforcing IBITI / USDT liquidity on PancakeSwap and other venues.
• Staking and community incentives – reward pools managed by the StakingModule and other incentive contracts.
• Team and contributors – subject to multi-year vesting schedules in contracts such as TeamVesting.
• Ecosystem and DAO programs – flexible allocations for future governance decisions, integrations and campaigns.

3. Smart-contract architecture

The IBITIcoin ecosystem is implemented as a suite of Solidity contracts. Each contract focuses on a narrow area: core token logic, fee calculation, staking, NFT utilities, DAO governance, treasury operations and so on. Most contracts are built on top of OpenZeppelin primitives such as ERC20, ERC721, Ownable, Pausable and ReentrancyGuard.

IBITIcoin.sol

The main BEP-20 token contract. It integrates with the FeeManager for trading fees and exposes hooks for external modules such as staking, DAO voting and treasury operations. The maximum supply of 100,000,000 IBITI is enforced at the contract level.

BaseToken.sol

Base token logic shared by IBITIcoin and potential future tokens in the ecosystem. Encapsulates common patterns for fee handling, interaction with routers and internal accounting.

FeeManager.sol

Central module for computing buy and sell fees. It combines base configuration with discounts and surcharges from several sources: staking status, VIP level, holding duration, whale size, NFT-based discounts and volatility coefficient. The final fee is clamped between 0% and 50%.

StakingModule.sol

Contract that allows users to stake IBITI for rewards. It tracks stakes, lockup periods, reward emissions and penalties for early exit. Staking status can be queried by the FeeManager to apply reduced trading fees.

NFTDiscount.sol

Utility contract that maps IBITINFT ownership and NFT attributes to fee discounts or other perks. Used by FeeManager and potentially other application contracts.

IBITINFT.sol

ERC-721 implementation of the official IBITI NFT collection. NFTs can carry discount tiers, access rights or cosmetic value inside the ecosystem.

NFTSaleManager.sol

Configurable sale engine for IBITINFT and future NFT collections. It supports phased sales, per-phase pricing, supply limits and optional vesting/lockups for NFTs or related rewards.

PhasedTokenSale.sol

Generic phased sale contract that can be reused for IBITI-based token sales. It is designed for transparent, on-chain public sales instead of opaque off-chain allocations. Not every phase has to be used; the contract is a tool, not a promise of specific sales.

TeamVesting.sol

Vesting contract for team and advisor allocations. Tokens are locked and released according to predefined cliffs and linear vesting schedules. Ownership of these contracts is separated from operational wallets.

DAOModule.sol / DAOModuleImplementation.sol

On-chain governance module with proposals, voting and execution hooks. In the early stages, it is used for non-critical and configuration-level actions. Over time, more powers can be delegated to the DAO as the system and community mature.

BuybackManager.sol

A module for performing on-chain buyback operations: swapping stablecoins or other assets for IBITI on PancakeSwap and optionally burning a portion of purchased tokens. It provides a transparent, scriptable mechanism for any future buyback-and-burn programs.

BridgeManager.sol

Manager for integrating IBITI with cross-chain bridges. It tracks allowed bridge endpoints and can coordinate mint/burn or lock/unlock flows as part of a bridged asset model. At the time of this version, it is a building block for future expansion rather than an active bridge.

UserStatusManager.sol

Registry of per-address flags (for example blocked, frozen, KYC-verified). Other contracts can consult this registry before processing transfers or sensitive operations. This allows compliance- or security-related policies to be enforced in a modular way.

VolumeWeightedOracle.sol & IUniswapV2Pair.sol

Components for observing on-chain trading data and extracting a volume-weighted measure of recent activity. The FeeManager can use this information to adjust fees during periods of very high volatility or abnormal volume.

AggregatorV3Interface.sol, ERC20Mock.sol and other auxiliary contracts

Auxiliary contracts used for testing, integration with price feeds and simulations. They do not hold user funds in production.

4. Trading-fee model (detailed)

The FeeManager contract encodes the logic for dynamic trading fees on IBITI. Instead of hard-coding a single flat percentage, the system allows multiple factors to increase or decrease the effective fee for a given trade, within strict global bounds.

4.1 Components of the fee

• Base buy/sell fee – the starting point for all trades. For example, base buy fee can be set low or zero to encourage entry, while base sell fee can be higher to support the treasury.
• Staking discount – addresses with active stakes in StakingModule may receive a percentage discount on selling fees, rewarding long-term commitment.
• VIP tiers – selected addresses or tiers can be marked as VIP, reducing fees for partners, key ecosystem participants or market makers.
• Holding duration – addresses that have held IBITI for a longer period (based on a holding tracker) can receive incremental discounts compared to very short-term addresses.
• Whale surcharge – large sell orders above configured thresholds can receive a positive adjustment to the fee to reduce sudden shocks to liquidity.
• NFT-based discounts – ownership of specific IBITINFT tokens or rarity tiers can reduce trading fees via the NFTDiscount module.
• Volatility coefficient – an optional multiplier derived from recent trading activity, such that fees can be temporarily higher during extreme volatility windows.

4.2 Hard limits and guarantees

All these factors are combined inside FeeManager and then clamped between a minimum and maximum fee. The contract explicitly enforces that the final fee cannot be negative and cannot exceed an upper bound (for example 50% of the trade amount). This prevents accidental or malicious configuration of unrealistically high fees.

All parameters (base fees, discount rates, whale thresholds, volatility settings) are stored on-chain and can be audited by anyone. Changes emit events and can later be governed by the DAO.

5. Staking & rewards

The StakingModule contract provides a way to lock IBITI and earn rewards over time. It is designed to be simple to understand: users deposit IBITI, wait, and claim rewards. All logic is on-chain; no one holds funds on behalf of users.

• Stakes are recorded per address with amount and optional lock parameters.
• Rewards are denominated in IBITI and can come from the treasury, protocol revenue or dedicated reward pools.
• Early withdrawal can be penalized according to parameters; penalties can be redirected to the treasury.
• Staked status can be used by FeeManager to reduce trading fees for long-term participants.

6. NFT layer & utilities

NFTs in the IBITI ecosystem are not just collectibles. Through IBITINFT and NFTDiscount, they can be mapped to concrete protocol-level utilities: fee discounts, access rights or other perks.

• IBITINFT defines one or more ERC-721 collections under the IBITI brand.
• NFTs can be sold through NFTSaleManager, accepting IBITI or other tokens depending on configuration.
• NFTDiscount links NFT ownership and attributes (such as rarity tiers) to fee-discount percentages or other benefits.
• Future dApps can read NFT ownership directly and build additional, independent utilities.

7. DAO governance

DAO functionality in IBITI is deployed via DAOModule and DAOModuleImplementation. It is intentionally scoped: the initial focus is on configuration changes and signaling, not on immediate control over all funds.

• Proposals – addresses that meet the threshold can create proposals with descriptions and encoded actions.
• Voting – IBITI balances (and potentially staked balances) determine voting power at a snapshot block.
• Execution – only whitelisted contracts and methods can be called via successful proposals, limiting blast radius.
• Evolution – over time, more parameters and modules can be migrated under DAO control as the system matures.

8. Liquidity & market structure

IBITI is initially traded on PancakeSwap V2 against USDT. The official pool is created and supported from an on-chain reserve wallet using scripted operations instead of manual, opaque moves.

• Primary pair – IBITI / USDT on PancakeSwap V2.
• Pair address – 0xADfb9F0f810311e9c01C27B380909A5FfC104Be0.
• Router – PancakeSwap V2 router at 0x10ED43C718714eb63d5aA57B78B54704E256024E.
• Stablecoin – USDT on BSC at 0x55d398326f99059fF775485246999027B3197955.

Liquidity operations (adding and, if needed, removing liquidity) are executed via Hardhat scripts that perform the full sequence: approve exact amounts to the router, add liquidity, then reset allowances back to zero. This reduces the attack surface compared to keeping large permanent allowances open.

9. Security considerations

IBITIcoin contracts are built using standard patterns and libraries, but no smart contract can be considered perfectly safe. The following measures are in place to reduce risk, without pretending to eliminate it completely.

• Use of OpenZeppelin – core primitives such as ERC20, ERC721, Ownable, Pausable and ReentrancyGuard are taken from widely audited libraries instead of being reimplemented from scratch.
• Pause mechanisms – critical modules (FeeManager, selected DAO hooks, treasury-related contracts) can be paused by the owner in case of emergencies or clear abuse. This is a last-resort tool, not a normal operational mode.
• No hidden upgradeable proxies – the system does not hide arbitrary upgradeability behind opaque proxy patterns. When a module is replaced, it happens via explicit deployment of a new contract and setting its address via public setter functions, visible on-chain.
• Separation of roles – operational wallets, treasury wallets, team vesting contracts and DAO contracts have distinct roles. A compromise of one does not automatically grant control over all others.
• Limited allowances – router and other external contracts are granted allowances on a per-operation basis and then reset back to zero, reducing the impact of a compromised router or integration.

Despite these measures, IBITIcoin remains an experimental DeFi protocol. Participants should expect that bugs, market events or external dependencies could lead to partial or total loss of funds. There is no implicit or explicit guarantee of capital protection.

10. Legal & regulatory note

IBITI does not represent equity, debt or any direct claim on a company or legal entity. It is a utility and governance token designed for use inside the IBITIcoin ecosystem.

• No ICO/IEO – IBITI has not been sold via centralized ICO, IEO or launchpad. Distribution is performed through on-chain liquidity, incentives and direct transfers.
• No guaranteed returns – holding IBITI does not entitle the holder to profits, dividends or any form of guaranteed return.
• Jurisdictional differences – regulation of digital assets differs between countries. Each participant is responsible for understanding and complying with local laws, including tax obligations.

11. How to buy IBITI (summary)

This is a short, practical summary of how to buy IBITI on BNB Smart Chain using PancakeSwap V2. Always verify contract addresses manually from official sources.

• Configure your wallet (for example MetaMask, Trust Wallet) for BNB Smart Chain (BSC, chainId 56).
• Fund your wallet with a small amount of BNB for gas fees and some USDT (BEP-20) for the swap.
• Open the official IBITI PancakeSwap link: https://pancakeswap.finance/swap?chain=bsc&outputCurrency;=0x47F2FFCb164b2EeCCfb7eC436Dfb3637a457B9bb&inputCurrency;=0x55d398326f99059fF77548524699

9027B3197955
• Connect your wallet and double-check the IBITI contract address before confirming the swap.
• Start with small test trades and pay attention to price impact and slippage settings.

12. Official links

• Website: https://www.ibiticoin.com
• Token contract: https://bscscan.com/token/0x47F2FFCb164b2EeCCfb7eC436Dfb3637a457B9bb
• PancakeSwap V2 (IBITI / USDT): https://pancakeswap.finance/swap?chain=bsc&outputCurrency;=0x47F2FFCb164b2EeCCfb7eC436Dfb3637a457B9bb&inputCurrency;=0x55d398326f99059fF775485246999027B3197955
• Telegram: https://t.me/IBITIcoin_chat
• X (Twitter): https://x.com/ibiticoin
• Facebook: https://www.facebook.com/ibiticoin.ibiticoin/
• LinkedIn: https://www.linkedin.com/in/ibiticoin
• GitHub: https://github.com/VOVAN1980/IBITIcoin