PZZL

WHITEPAPER · VERSION 1.0

PZZL Token

The BEP-20 utility token behind Bitcoin Puzzle — rewards, deposits, and bridge-based accounting for a puzzle app built on Bitcoin address mechanics.

BNB Smart Chain | BEP-20 | EIP-2612 Permit | Fixed Supply · No Mint

July 2026

bitcoin.xomuak.com/token.html

0xC5262cdB7404aB7e232A1B5cA4c3e25250B716E7

BEFORE YOU READ ON

Important Notice

NATURE OF THIS DOCUMENT

This whitepaper describes PZZL, a BEP-20 utility token on BNB Smart Chain, and the Bitcoin Puzzle app it is used in. It is published for informational purposes to explain how the token works, how it is used, and the direction the project intends to take. It is not a prospectus, offering document, or solicitation to buy or sell any asset in any jurisdiction.

NOT INVESTMENT ADVICE

Nothing in this document is financial, investment, legal, or tax advice. PZZL is a utility token intended for use inside the Bitcoin Puzzle app, not an investment product, and it does not represent equity, a share of profits, or a claim on any entity. Anyone considering acquiring or using PZZL should do their own research and consult independent advisors as needed.

NO GUARANTEES

Cryptocurrency and utility tokens carry risk, including price volatility, low or variable liquidity, smart contract risk, and regulatory uncertainty that varies by jurisdiction. The project makes no promise or guarantee of value, future price, liquidity, or exchange listing for PZZL. Roadmap items in Section 8 describe intended direction, not commitments with fixed dates.

FORWARD-LOOKING STATEMENTS

Statements about future features, integrations, or milestones are forward-looking and subject to change based on development progress, technical constraints, and market conditions. This document may be revised; the current version is always available at the project's official token page.

CONTENTS

Table of Contents

SECTION 01

Introduction

Bitcoin Puzzle is a mobile puzzle app in which players work with Bitcoin address mechanics through an interactive game interface. PZZL is the BEP-20 utility token that sits underneath the app's economy: it is used for in-app rewards, for depositing and withdrawing value between a player's wallet and the app, and for bridge-based accounting that keeps on-chain holdings and in-app balances in sync.

PZZL is deployed on BNB Smart Chain as a standard, ERC-20-compatible BEP-20 token, so it works with any compatible wallet or exchange without requiring proprietary infrastructure to hold or move. This document is the reference for anyone who wants to understand PZZL beyond the summary on the official token page: what the contract does, how supply is fixed, how the token is used inside the app, and where the project is headed.

How this document is organized

Sections 3–5 cover the token itself — its specifications, supply mechanics, and the smart contract logic that enforces them. Sections 6–7 cover how PZZL is actually used inside

Bitcoin Puzzle. Section 8 lays out the project's roadmap, and Sections 9–11 cover who is behind the project and how to get in touch or verify any claim made here directly on-chain.

> Every technical claim in this document can be independently verified by reading the published contract source or inspecting it on BscScan — nothing here depends on trusting a claim that can't be checked on-chain.

SECTION 02

Purpose & Vision

Bitcoin Puzzle's premise is that Bitcoin's underlying mechanics — addresses, keys, balances — are more approachable when experienced through play rather than through documentation. PZZL exists to give that experience a real, freely transferable on-chain asset rather than a closed, in-app-only point system.

What PZZL is designed to do

- **Reward engagement.** Players earn PZZL for activity and progress inside the app, giving in-app achievement a real on-chain counterpart.
- **Provide a deposit and withdrawal rail.** PZZL can move from a player's wallet into their in-app balance, and back out again, through the bridge contract.
- **Keep accounting consistent.** The bridge reconciles on-chain holdings with in-app state, so a player's wallet and their in-game balance don't drift apart.
- **Remain a standard, portable asset.** PZZL is a plain BEP-20 token with no special permissions reserved for the issuer — it behaves the same way inside any wallet or exchange as it does inside Bitcoin Puzzle.

Design philosophy

The token contract favors simplicity and predictability over flexibility. There is no admin key that can mint new supply, freeze an account, or change how the token behaves after deployment. That constraint is intentional: it means the rules described in this document are the rules enforced by the contract, permanently, not policy that could be altered later.

SECTION 03

Token Overview

Token name	**PZZL**
Symbol	**PZZL**
Standard	**BEP-20 (ERC-20 compatible) with EIP-2612 Permit**
Network	**BNB Smart Chain Mainnet**
Chain ID	**56**
Decimals	**18**
Total supply	**10,000,000,000 PZZL**
Max supply	**10,000,000,000 PZZL — hard-capped, no mint function exists**
Emission	**None — fully minted at deployment, no inflation**
Contract address	`0xC5262cdB7404aB7e232A1B5cA4c3e25250B716E7`
Bridge contract	`0x4B32D5eE37AF41a49EDE0edafC00Db7498Ec5BeB`

All figures above can be verified directly against the deployed bytecode and the constants defined in the published Solidity source — none of them depend on documentation staying in sync with the contract.

SECTION 04

Tokenomics

PZZL uses the simplest supply model available to an ERC-20/BEP-20 token: a single fixed mint at deployment, and nothing else.

Supply mechanics

- **One-time mint.** The full 10,000,000,000 PZZL supply was created in a single transaction at contract deployment, credited to the deploying address, with a standard `Transfer` event emitted from the zero address.

- **No further issuance.** The contract does not expose a mint function of any kind. Supply cannot be increased after deployment — not by the original deployer, not by any other address, under any circumstance.
- **No burn mechanism.** The contract also does not expose a burn function, so total supply cannot be deliberately reduced on-chain either. Total supply, once set, is permanent.
- **No admin override.** There is no owner role, pause switch, or upgrade mechanism that could alter these rules later. The supply model in this document is the supply model enforced by the bytecode, indefinitely.

Circulating supply

Because every PZZL balance is a public entry on BNB Smart Chain, circulating supply — the portion of the 10 billion total that is actively held and transferable outside of any reserve or treasury wallet — can be tracked live by anyone through the contract's holder list on BscScan, rather than relying on a static figure in this document that could go stale.

TOTAL / MAX SUPPLY	CIRCULATING SUPPLY
10,000,000,000 PZZL	**Live on BscScan**

Allocation

This document does not publish a fixed team / liquidity / marketing allocation breakdown at this time. Because the contract itself encodes no vesting or lock-up schedule, any allocation or distribution commitments the project makes will be published and updated through the official channels listed in Section 11, and can be cross-checked against on-chain wallet activity at any time.

SECTION 05

Smart Contract Architecture

PZZLToken is written directly against the ERC-20 and EIP-2612 interfaces rather than an inherited framework, keeping the deployed logic compact and easy to audit line by line.

Core ERC-20 / BEP-20 interface

Standard `transfer`, `approve`, `transferFrom`, `allowance`, `balanceOf`, and `totalSupply`, backed by ordinary Solidity mappings for balances and allowances. Failure conditions use custom errors (`InsufficientBalance`, `InsufficientAllowance`, `TransferToZeroAddress`, `ApproveToZeroAddress`) instead of require-string reverts, which reduces deployed bytecode size and gas cost on failure.

Allowance safety helpers

`increaseAllowance` and `decreaseAllowance` let a holder adjust an existing approval by a delta rather than overwriting it outright, which avoids the classic ERC-20 front-running race condition that a plain `approve()` call is exposed to. `decreaseAllowance` reverts with `AllowanceBelowZero` rather than underflowing.

EIP-2612 Permit

PZZL supports gasless, signature-based approvals via `permit()`: a holder signs an EIP-712 typed message off-chain, and any relayer can submit it on-chain to set an allowance without the holder needing to send a transaction or hold BNB for gas at that moment. The domain separator is computed once at deployment and cached, then automatically rebuilt if the chain ID ever changes — a safeguard against chain-fork replay of signed permits.

Permit signatures are also checked against the EIP-2 malleability bound: values of `s` above the curve's half-order, or a `v` value other than 27/28, are rejected outright with `InvalidSignature`, closing off a known class of signature-malleability issues in naive `ecrecover` usage.

No privileged access

There is no owner address, admin role, pause function, blacklist, or proxy/upgrade pattern anywhere in the contract. Once deployed, PZZL's behavior cannot be modified by the project team or anyone else — what is described in this whitepaper is enforced by immutable, deployed bytecode.

SECTION 06

Token Utility in Bitcoin Puzzle

Inside the Bitcoin Puzzle app, PZZL is the unit the app's economy is built around, rather than a token bolted on after the fact. Its core uses are:

1. **In-app rewards.** Players earn PZZL for activity and progress within Bitcoin Puzzle's puzzle mechanics.
2. **Deposits.** Players can move PZZL from their own wallet into their in-app balance to fund gameplay.
3. **Bridge-based accounting.** The bridge contract reconciles a player's on-chain PZZL activity with their in-app balance, so the two stay consistent rather than drifting apart.
4. **Withdrawals.** Balances accrued inside the app can be moved back out to the player's own wallet as ordinary, freely transferable on-chain PZZL.

Because PZZL is a standard BEP-20 asset, none of this locks a player in: at any point, PZZL sitting in a wallet is a normal token that behaves like any other BEP-20 asset in any compatible wallet or exchange, independent of the app.

SECTION 07

Bridge & Cross-Balance Accounting

The bridge contract at `0x4B32D5eE37AF41a49EDE0edafC00Db7498Ec5BeB` is the component that connects a player's wallet-held PZZL to their in-app balance. Conceptually, it plays a similar role to a deposit/withdrawal contract in any token-based application: it observes PZZL moved into it as deposits, and releases PZZL back out as withdrawals, giving the app an on-chain-anchored source of truth for a player's balance rather than a purely off-chain ledger.

The project's shared contract library defines a `SafeERC20` helper used across PZZL's contracts — including transfer paths the bridge relies on — to guard against a transfer silently failing or being sent to a non-contract address. In practice, this means a transfer call is checked for both a successful call and, where the target token returns a boolean, a truthy result, rather than assuming success whenever a call doesn't revert.

> The bridge contract's own source is published separately from the token contract; readers who want function-level detail on deposit/withdrawal logic should refer to the bridge contract on BscScan directly rather than this summary.

SECTION 08

Roadmap

The roadmap below describes intended direction, not fixed commitments. Phases may be reordered, combined, or adjusted as development and community needs evolve.

Phase 1 — Foundation Completed

PZZL contract deployed and verified on BscScan, bridge contract live, official token page and app links published.

Phase 2 — App Integration In progress

Deposits and withdrawals between wallet and in-app balance via the bridge contract, wallet-connect flow inside Bitcoin Puzzle, and expanded reward mechanics for active players.

Phase 3 — Ecosystem Growth

Growing the Telegram community, this whitepaper's continued expansion, additional listings and liquidity venues, and player feedback shaping new in-app reward types.

Phase 4 — Expansion

Broader BEP-20 ecosystem integrations, exploration of additional utility for PZZL beyond the current app, and continued open development of the contract suite.

SECTION 09

Team & Development

Bitcoin Puzzle and the PZZL token are built and maintained by a single independent developer. Development activity, including the contracts referenced throughout this document, is publicly visible on GitHub.

GitHub	**github.com/exxuslee**
LinkedIn	**linkedin.com/in/exxus**

SECTION 10

Risk Factors

Anyone acquiring or using PZZL should understand the following risk categories before doing so:

- **Market risk.** Like any freely traded token, PZZL's market price, if and where it trades, can be highly volatile and may fall to zero. Past activity is not an indicator of future value.
- **Liquidity risk.** There is no guarantee of deep, continuous liquidity on any exchange for PZZL at any point in time.
- **Smart contract risk.** Although the token contract is intentionally simple and has no admin privileges, no smart contract can be guaranteed entirely free of unforeseen issues. Readers are encouraged to review the published source directly.
- **Bridge risk.** The bridge contract is a separate piece of infrastructure from the token itself; its correct operation is required for deposits and withdrawals between wallet and app to function as described.
- **Regulatory risk.** The legal treatment of utility tokens varies by jurisdiction and can change. It is each individual's responsibility to determine whether acquiring or using PZZL is permitted where they live.
- **Single-developer risk.** Bitcoin Puzzle and PZZL are currently developed and maintained by one person rather than a distributed team or foundation, which concentrates key-person and continuity risk.

- **Early-stage risk.** The app and its token utility are under active development; features described as in-progress or planned in Section 8 may change materially before release.

This list is not exhaustive. It is provided to inform judgment, not to replace it — evaluate PZZL the same way you would evaluate any other early-stage token before acquiring or using it.

SECTION 11

Contact & Resources

All claims in this document can be cross-checked directly against the resources below.

Official token page	**bitcoin.xomuak.com/token.html**
Token explorer	**bscscan.com/token/0xC5262c...B716E7**
Telegram	**t.me/btc_pzzl**
Email	**exxlexxlee@gmail.com**
GitHub	**github.com/exxuslee**
Official app	**Bitcoin Puzzle — Google Play**

PZZL Token Whitepaper · Version 1.0 · July 2026 · This document may be revised — the latest version is always linked from the official token page.