GetLuck
Web3 luck games

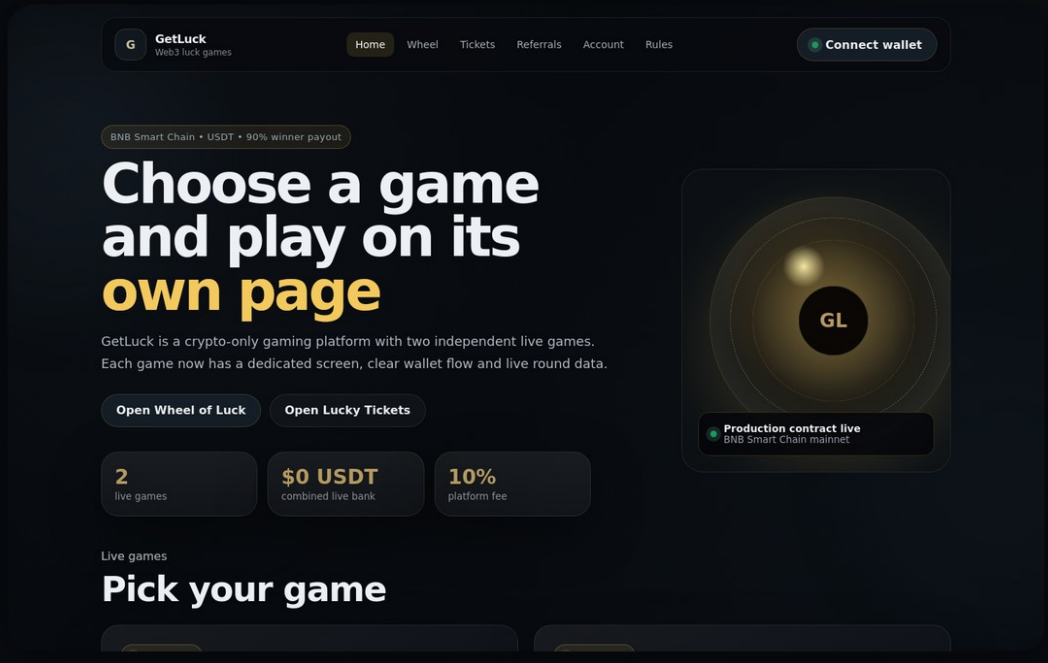
PUBLIC TECHNICAL DOCUMENT

Technical White Paper

Production Web3 gaming platform on BNB Smart Chain

Transparency note

GetLuck does not issue a proprietary token. Participation uses the existing USDT contract on BNB Smart Chain. This document is a project white paper, not an independent security audit.



Version 1.0 | 16 August 2026

https://getluck.pro

BNB Smart Chain Mainnet

Executive summary

GetLuck is a production Web3 gaming application for two transparent, crypto-only luck games: Wheel of Luck and Lucky Tickets. Users authenticate with an EVM wallet, approve only the amount of USDT required for a chosen action, submit the game transaction to the production contract, and track the confirmed result through the application.

Important: GetLuck does not issue or sell a proprietary token. Game participation uses the existing USDT payment token on BNB Smart Chain. There is no token sale, staking yield, deposit interest, or promised investment return.

The production contract is deployed on BNB Smart Chain Mainnet. Project evidence records its source as verified on BscScan with Exact Match and records ownership verification of the contract address. The platform fee is 10% of each successfully completed round bank; 90% is paid to the winning wallet. Referral rewards, when applicable, are funded from the platform's fee rather than added on top of the round bank.

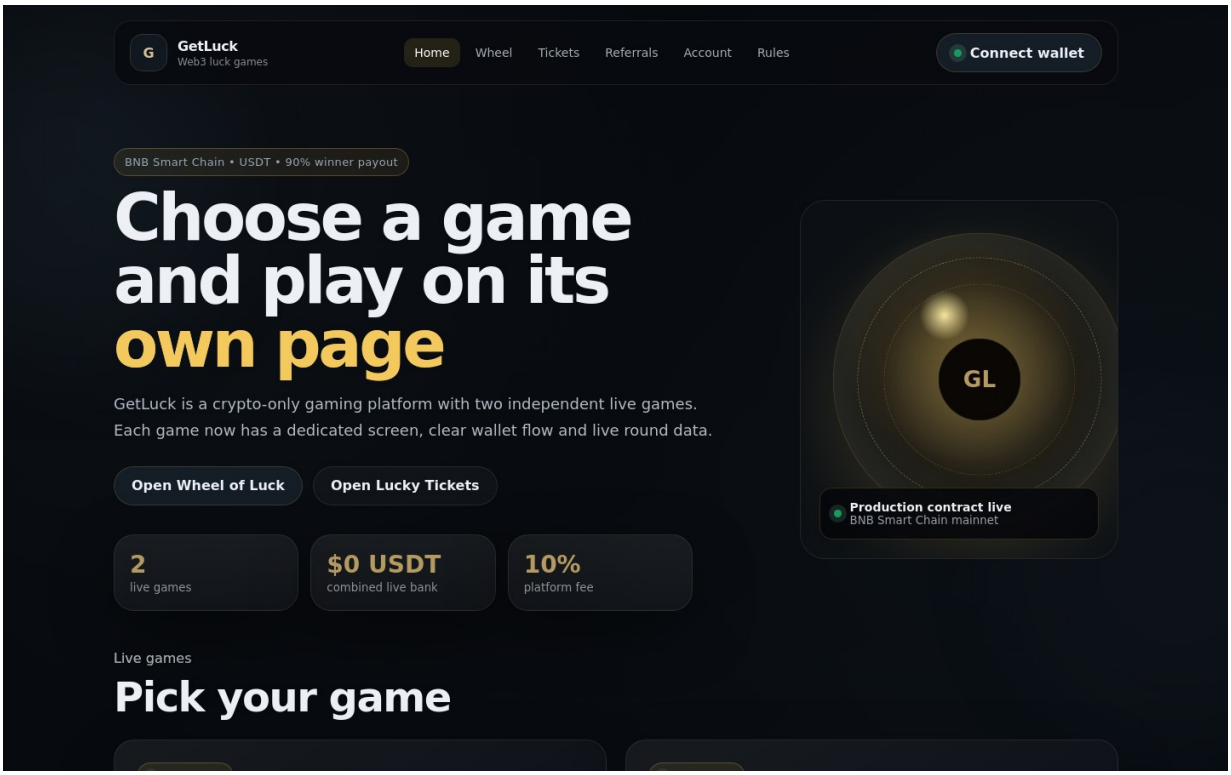
Reviewer quick facts

Item	Production value
Production website	https://getluck.pro
Network	BNB Smart Chain Mainnet
Chain ID	56 (0x38)
Payment token	USDT / Binance-Peg BSC-USD
Payment token address	`0x55d398326f99059fF775485246999027B3197955`
Production game contract	`0xb60E3302F0c30F0C2CD4F534762569880d342Baa`
Owner / deployer / fee wallet	`0x3a4b72EA96366013e231924f8fC9e69B215450f6`
Platform fee	10% (1000 basis points)
Winner payout	90% of completed round bank
Games	Wheel of Luck; Lucky Tickets
Native GetLuck token	None
Classic registration	None - wallet-based access
Public UI language	English
Age notice	18+

Document purpose. This white paper describes the current GetLuck product, its production deployment, user transaction flow, game economics, round-integrity model, operational architecture, and disclosed risks. It is prepared for public technical transparency and third-party reputation/security review.

Contents

1. Product scope and principles
2. Wallet authentication and transaction flow
3. Game mechanics
4. Round lifecycle and winner-selection integrity
5. Economics and referral model
6. Smart contract and production deployment
7. Backend, indexing and operational architecture
8. Security controls and trust model
9. Production evidence and public verification
10. Risks, responsible use and disclosures
Appendix A. Public addresses and transactions
Appendix B. Verification references



Current production interface after the August 2026 multi-page UI acceptance.

1. Product scope and principles

GetLuck is an entertainment-oriented Web3 application. The product is deliberately narrow: the production version contains two games, a wallet-based user area, referral accounting, public rules, and an owner-facing administration layer. It does not use bank cards, bank transfers, Stripe, YuKassa, or other fiat payment rails.

Core product principles

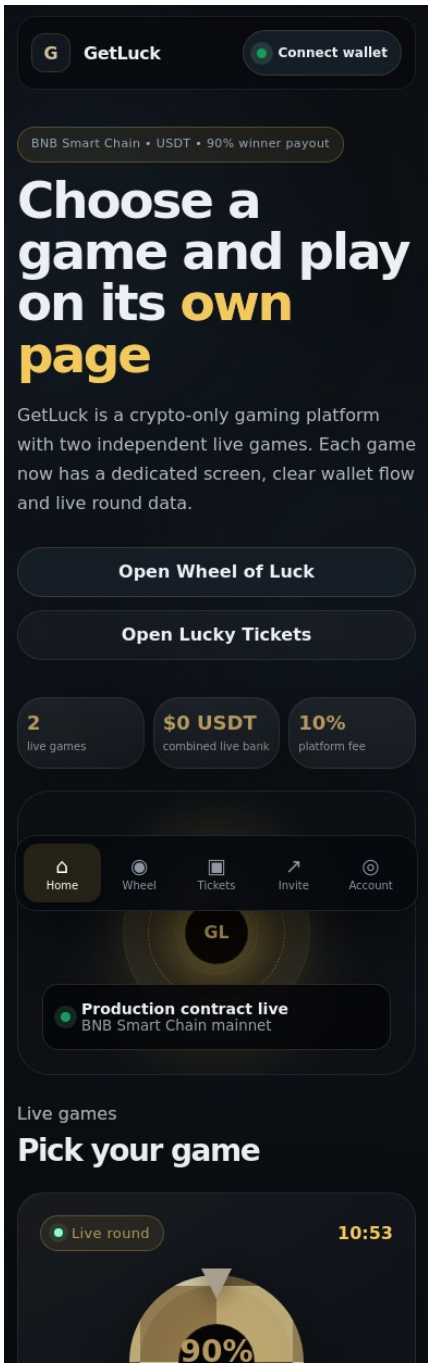
- Crypto-only participation through a user-controlled EVM wallet.
- No username/password/email registration is required for basic access; the wallet address is the primary user identifier.
- No proprietary GetLuck token is minted, sold, staked, or marketed.
- Each game has an explicit round bank, end time, fee and payout rule.
- The player signs each blockchain action; the application does not silently initiate token transfers.
- Public interfaces separate Home, Wheel, Tickets, Referrals, Account and Rules into dedicated pages.
- The owner/admin interface is not exposed in the ordinary public navigation.

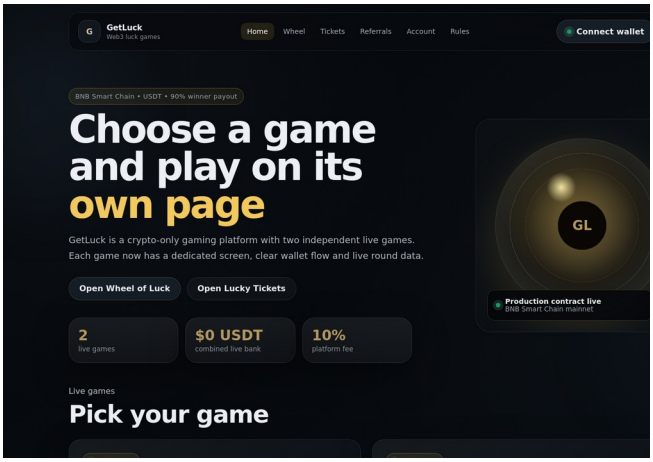
What GetLuck is not

- It is not a token sale, ICO, presale, staking product, lending protocol, investment fund, or yield product.
- It does not promise a fixed or guaranteed return. A user can lose the full amount contributed to a game round.
- It does not require a user to send funds manually to an administrator.
- It does not use multi-level referral economics or a downline investment structure in the current MVP.

Production interface

The public site uses a dark, restrained Web3/gaming visual style. The live UI displays network and contract status, game bank, round timing, transaction steps, user history and referral statistics without exposing raw backend data or private operational credentials.





Representative mobile and desktop production views.

2. Wallet authentication and transaction flow

GetLuck uses wallet-based authentication and on-chain authorization. Connecting a wallet and authenticating to the application are separate from spending. Authentication is performed with a free message signature; the message-signing step does not transfer tokens.

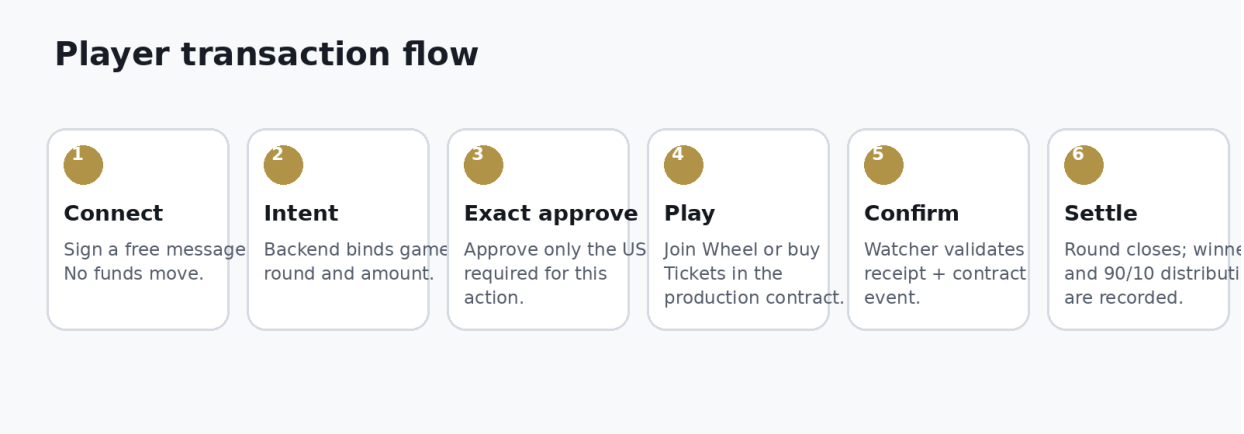


Figure 1. Player transaction flow.

Exact-amount approval

When USDT allowance is insufficient, the production UI requests an approval for the amount required by the selected action rather than requesting an unlimited allowance. The user then separately confirms the game transaction. Controlled production QA has demonstrated exact-amount approval behavior and zero residual allowance after the tested 5 USDT ticket purchase.

Network and token checks

- Expected network: BNB Smart Chain Mainnet, chain ID 56.
- Expected payment token: 0x55d398326f99059fF775485246999027B3197955.
- User actions are rejected or paused when the connected wallet is on the wrong network or a transaction is not confirmed.
- Only confirmed contract interactions are added to the application read model and round history.

Wallet compatibility

MetaMask is the primary tested wallet flow. The architecture is EVM-compatible and can support additional wallets through compatible connection mechanisms. Wallet availability may depend on browser, mobile wallet environment, and third-party wallet infrastructure.

3. Game mechanics

Wheel of Luck

Wheel of Luck is a contribution-weighted pool. A confirmed USDT contribution becomes part of the current round bank. A wallet's winning probability is proportional to that wallet's confirmed contribution relative to the confirmed bank.

Wheel parameter	Production rule
Minimum entry	1 USDT
Current production round target	Approximately 15 minutes
Repeated entries	Supported; confirmed contributions accumulate for the wallet
Chance model	Wallet contribution / confirmed round bank
Winner payout	90% of completed round bank
Platform fee	10% of completed round bank

Lucky Tickets

Lucky Tickets is a ticket-based round. Each confirmed ticket has equal probability. A wallet may buy one or multiple tickets in a single transaction; ownership of the selected winning ticket determines the winning wallet.

Ticket parameter	Production rule
Ticket price	5 USDT
Current production round target	Approximately 6 hours
Quantity	One or more tickets per purchase
Chance model	Equal chance per confirmed ticket
Winner payout	90% of completed round bank
Platform fee	10% of completed round bank

Confirmed-data rule

Unconfirmed, rejected or failed wallet transactions are not treated as game participation. The displayed bank and player history are based on confirmed contract events processed by the blockchain indexing pipeline.

4. Round lifecycle and winner-selection integrity

The round lifecycle is automated. Operational services create and synchronize rounds, stop accepting new participation when a round closes, settle completed rounds and support explicit cancel/refund recovery paths for technical failure conditions.

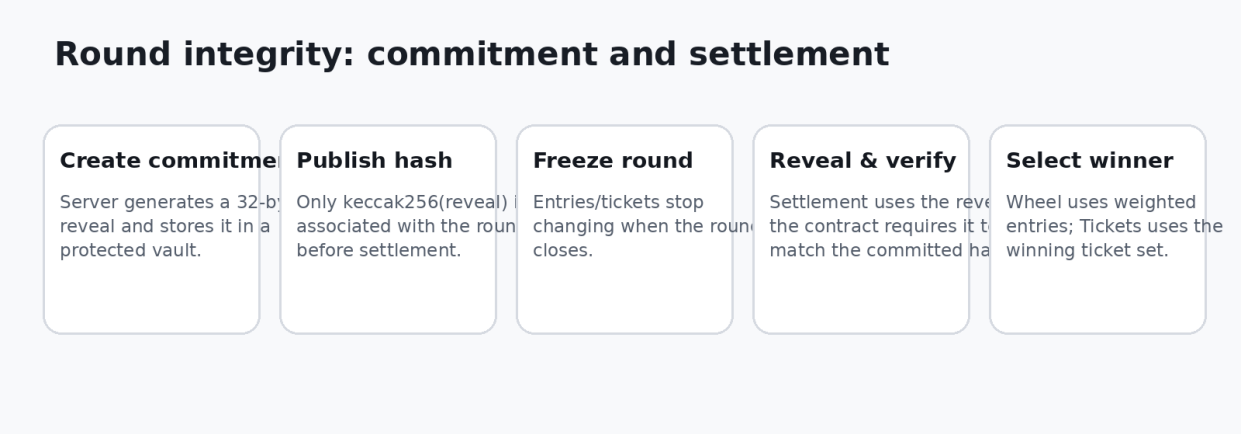


Figure 2. Commitment/reveal round-integrity model.

Commitment / reveal model

The production architecture uses a server-side 32-byte seed reveal and a keccak256(bytes32) commitment that is associated with the round. The reveal is stored in a protected server-side vault and is not exposed in public game-state responses. At settlement, the reveal must match the previously committed hash.

This construction prevents changing a committed reveal after the commitment is established. It is not presented as an external VRF or as a trustless randomness oracle. The current MVP retains an owner-operated settlement component, which is disclosed as an operational trust point. A future migration to an external VRF remains a possible hardening path.

Selection rules

- Wheel of Luck: winner selection is weighted by the confirmed contribution shares fixed for the closed round.
- Lucky Tickets: the winner is the owner of the selected confirmed ticket from the fixed ticket set.
- Selection occurs only after the round has stopped accepting entries.
- Round ID, bank, entries/tickets, commitment and settlement transaction references are retained for auditability.

Recovery paths

The production contract exposes explicit lifecycle operations including round creation, activation/availability, finish, cancellation and refund of cancelled rounds. The server-side orchestrator prepares and reconciles these actions while protecting against duplicate execution.

5. Economics and referral model

Round economics

Component	Rule
Winning wallet	90% of completed round bank
Platform	10% of completed round bank
Payment asset	Same USDT token used for participation

Referral accounting

The referral program is direct and wallet-based. An invited wallet can be bound to one referrer subject to self-referral and duplicate-binding protections. Referral rewards are calculated from the platform's 10% fee, not as an additional deduction from the 90% winner payout and not as an extra charge on top of the round bank.

The referral percentage is configurable. The MVP records invited-wallet turnover, accrued reward, paid reward and pending reward. Current referral settlement uses controlled crypto payout rather than an automatic multi-recipient on-chain split for every game transaction.

No investment-yield model

Referral rewards are tied to confirmed activity of directly invited wallets. GetLuck does not advertise referral rewards as investment yield, does not promise a return on deposited capital, and does not implement a multi-level downline payout structure in the current production scope.

6. Smart contract and production deployment

The production game contract is GetLuckGameEscrow on BNB Smart Chain Mainnet. The project evidence package records the deployed bytecode, expected owner, fee wallet, payment token and fee basis points as matching the production configuration.

Contract field	Production value
Contract name	GetLuckGameEscrow
Network	BNB Smart Chain Mainnet
Chain ID	56
Contract address	`0xb60E3302F0c30F0C2CD4F534762569880d342Baa`
Deploy transaction	`0xc728bfc1061675acc860bc1387f46a74d53cf22d171b580dbfbd2b11e3362592`
Payment token	`0x55d398326f99059fF775485246999027B3197955`
Owner / fee wallet	`0x3a4b72EA96366013e231924f8fC9e69B215450f6`
Fee	1000 bps (10%)
Source verification	BscScan - Source Code Verified, Exact Match
Compiler	Solidity v0.8.24+commit.e11b9ed9
Optimizer	Enabled, 200 runs
viaIR	true
EVM target	Shanghai
License	MIT

Core contract responsibilities

- Accept ERC-20 participation for the configured payment token.
- Maintain round metadata and participant/ticket entries.
- Support createRound, activateRound, joinWheel, buyTickets, finishRound, cancelRound and refundCancelledRound lifecycle operations.
- Validate the committed seed reveal at settlement.
- Apply the configured 90/10 round distribution.
- Emit on-chain events that the backend watcher can independently index.

Open verified production contract on BscScan

7. Backend, indexing and operational architecture

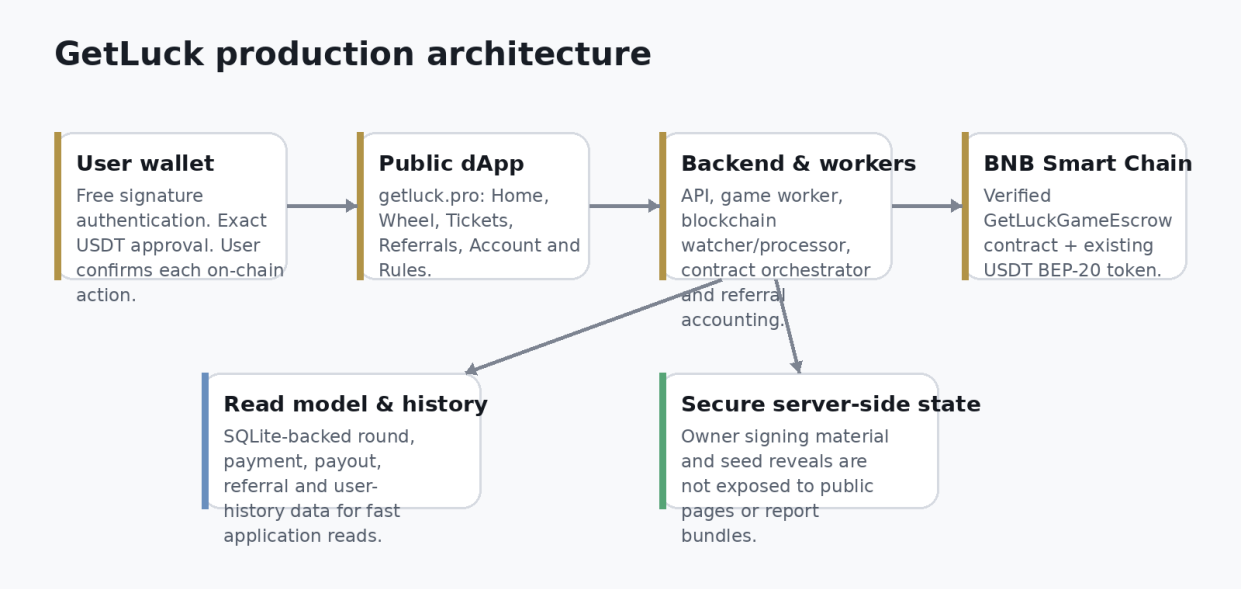


Figure 3. High-level production architecture.

GetLuck uses a hybrid architecture: blockchain transactions are the source of truth for participation and settlement, while the backend maintains a fast read model for the user interface, account history, referral accounting and operational recovery.

Production service roles

Component	Responsibility
Application/API	Serves the public site and wallet/account APIs.
Game worker	Maintains local round lifecycle and game-state preparation.
Blockchain watcher	Reads BNB Smart Chain events and receipts.
Blockchain processor	Applies confirmed blockchain effects to the read model with duplicate protection.
Contract orchestrator	Coordinates owner-side lifecycle actions such as create/finish/cancel/refund.
Referral worker	Reconciles referral accounting from confirmed participation.
nginx	Terminates HTTPS and routes public requests to the application.

Read-model resilience

The application uses SQLite for production read-model and history data. After an observed SQLITE_BUSY condition during concurrent reads/writes, the database wrappers were hardened with a 5-second SQLite busy timeout and bounded retry behavior. Subsequent production acceptance completed 80/80 API probes without HTTP 4xx/5xx responses or new lock errors.

This is an application-availability control; it does not alter blockchain settlement semantics or token balances.

8. Security controls and trust model

GetLuck's MVP security objective is to prevent obvious fund-accounting errors, duplicate transaction effects, duplicate settlement, unintended token allowances, public exposure of signing secrets and silent manual changes to completed game results.

Control	Purpose
Exact token allowance	The UI requests the required USDT amount instead of an unlimited approval.
Chain/token validation	The expected network and payment-token contract are fixed in the production configuration.
Receipt/event validation	Participation enters the read model only after blockchain confirmation and expected contract evidence.
Transaction deduplication	A confirmed blockchain transaction is not counted twice.
Round freeze	New participation cannot be added to a round after its closing state.
Settlement idempotency	Lifecycle state and payout records protect against repeated settlement effects.
Commit/reveal integrity	The reveal must match the commitment associated with the round.
Protected seed vault	Seed reveals are stored outside public responses in a root-controlled server-side database.
Private-key isolation	Owner signing material is stored server-side and is not embedded in public JavaScript, HTML or reports.
Admin separation	Owner/admin functions are kept outside ordinary public navigation and protected by owner authentication.
Secret-safe reporting	Operational report bundles are designed to exclude private keys, seed phrases and protected RPC credentials.

Disclosed trust points

- The current MVP has an owner/deployer role and server-operated lifecycle/orchestration components; it is not claimed to be fully decentralized.
- The current randomness design is commitment/reveal based and is not claimed to be a third-party VRF.
- BNB gas is required for owner lifecycle transactions; insufficient gas can temporarily prevent round creation or settlement until the wallet is funded.
- The project has not represented this white paper as an independent third-party smart-contract security audit.

9. Production evidence and public verification

GetLuck maintains public blockchain evidence that can be reviewed independently of the website. The production contract address, deployment transaction and representative game transactions are listed below. Project evidence also records BscScan source verification as Exact Match and successful ownership verification of the production contract address.

Representative verified production events

Evidence	Transaction	Purpose
Contract deployment	0xc728bfc1...e3362592	Deployment of the production GetLuckGameEscrow contract.
Wheel participation	0xb6499333...8c1e4b30	Known successful 5 USDT WheelJoined interaction.
Exact USDT approval	0xb32e5019...39189023	Controlled exact-amount approval used in production QA.
Lucky Tickets purchase	0x72c28891...16a6f5ca	Confirmed production ticket purchase.
Settlement / payout	0xebb04753...c6f32036	Confirmed settlement/payout evidence from production testing.

Source verification

The production source was published to BscScan as GetLuckGameEscrow with Source Code Verified / Exact Match. Project records identify compiler v0.8.24, optimizer enabled with 200 runs, viaIR enabled, Shanghai EVM target and MIT license.

BscScan source-code page

The project also completed BscScan address-ownership verification for the production contract. No reviewer should need access to a private key, seed phrase, wallet password or protected RPC credential to verify the public deployment.

Reputation-review context

A wallet/security-provider warning is currently subject to external review. GetLuck has supplied developer-verification information and is providing this white paper in response to a request for project documentation. The project does not represent the warning as resolved until the external provider completes its own review.

10. Risks, responsible use and disclosures

Web3 games involve financial and technical risk. This section is intentionally direct so that users, reviewers and integrators can distinguish the product's mechanics from guaranteed-return or investment claims.

Risk	Disclosure
Loss of game entry	Only one winning outcome exists per round; non-winning entries do not receive the winner payout.
Blockchain finality	Confirmed blockchain transactions are generally irreversible.
Smart-contract risk	A defect in deployed contract logic can affect funds or game operation. This document is not an independent audit.
Stablecoin/token risk	USDT/BSC-USD is an external token whose issuer, bridge and market risks are outside GetLuck's control.
Network/gas risk	BNB Smart Chain congestion, RPC availability or insufficient owner gas can delay lifecycle operations.
Wallet risk	Users are responsible for securing their wallet, seed phrase and device. GetLuck never needs a user's seed phrase.
Operational trust	The MVP has an owner role and server-side orchestrator; it is not fully decentralized.
Randomness/liveness	Commitment/reveal limits post-commit changes but is not an external VRF and includes an operator liveness dependency.
Regulatory/local-law risk	Users are responsible for determining whether participation is lawful in their jurisdiction.

Responsible-use position

- The public interface is marked 18+.
- Users should interact only with the published production domain and verify the contract/token addresses before signing.
- GetLuck does not ask users to disclose seed phrases or private keys.
- Users should read the on-site Rules page and understand that participation can result in losing the full game entry.

White-paper disclaimer. This document describes the project and its technical controls. It is not legal advice, financial advice, a promise of profit, a certification of regulatory status, or an independent security audit.

Appendix A. Public addresses and transactions

Public item	Value
Website	https://getluck.pro
Rules	https://getluck.pro/rules
BNB Smart Chain	Chain ID 56 / 0x38
Payment token	`0x55d398326f99059fF775485246999027B3197955`
Production contract	`0xb60E3302F0c30F0C2CD4F534762569880d342Baa`
Owner / deployer / fee wallet	`0x3a4b72EA96366013e231924f8fC9e69B215450f6`

Representative transactions

Contract deployment: 0xc728bfc1061675acc860bc1387f46a74d53cf22d171b580dbfbd2b11e3362592

Deployment of the production GetLuckGameEscrow contract.

Wheel participation: 0xb649933325fcd15059e6f3d4278ec6a58564ce4d43c276e691a258638c1e4b30

Known successful 5 USDT WheelJoined interaction.

Exact USDT approval: 0xb32e5019121bade29acb89dd9b37378c31511308ec024fc74ab3de1d39189023

Controlled exact-amount approval used in production QA.

Lucky Tickets purchase: 0x72c28891f0e5081b3ba8df75251add05fd947b30defcc68b55fbe5d116a6f5ca

Confirmed production ticket purchase.

Settlement / payout: 0xebb047538386a939c9001e44539ba093d86fa97141ed972ea2a93c24c6f32036

Confirmed settlement/payout evidence from production testing.

Additional production round-creation evidence

Lucky Tickets round creation: 0xe5482249c0e5f0e62fd1f8f37014d138f66d98127f6af016087b4de2fc10efd6

Wheel of Luck round creation:
0xb091975630968eb6c54afadc9b77e46d7398da20412180d8a27239edcfe66667

Appendix B. Verification references

GetLuck production website

https://getluck.pro

GetLuck production rules

https://getluck.pro/rules

GetLuck contract source on BscScan

https://bscscan.com/address/0xb60E3302F0c30F0C2CD4F534762569880d342Baa#code

USDT / BSC-USD token on BscScan

https://bscscan.com/token/0x55d398326f99059fF775485246999027B3197955

BNB Chain wallet configuration (Chain ID 56)

https://docs.bnbchain.org/bnb-smart-chain/developers/wallet-configuration/

BNB Chain JSON-RPC documentation

https://docs.bnbchain.org/bnb-smart-chain/developers/json_rpc/json-rpc-endpoint/

Document provenance

This white paper consolidates the project's approved technical specification and production acceptance evidence through August 2026. Where the original specification offered multiple implementation options (for example network choice or future VRF), this document uses the subsequently deployed production configuration as the authoritative current state.

Version history

Version	Date	Description
1.0	16 August 2026	Initial public technical white paper for the production GetLuck deployment.

GetLuck

https://getluck.pro

End of document