ROBOTBULLS

A Decentralized Platform for Trading Robots

Abstract

RobotBulls is a decentralized, open-source protocol for publishing and executing algorithmic trading strategies (“robots”) directly on-chain. The system enables developers to contribute trading algorithms, users to independently deploy them through smart contracts, and governance participants to coordinate protocol upgrades and parameters.

All operations are executed by autonomous smart contracts on blockchain infrastructure, with code and metadata stored in decentralized storage systems. Governance is managed by a community-driven DAO, supported by the RobotBulls Token (RBT), which functions solely as a governance coordination tool.

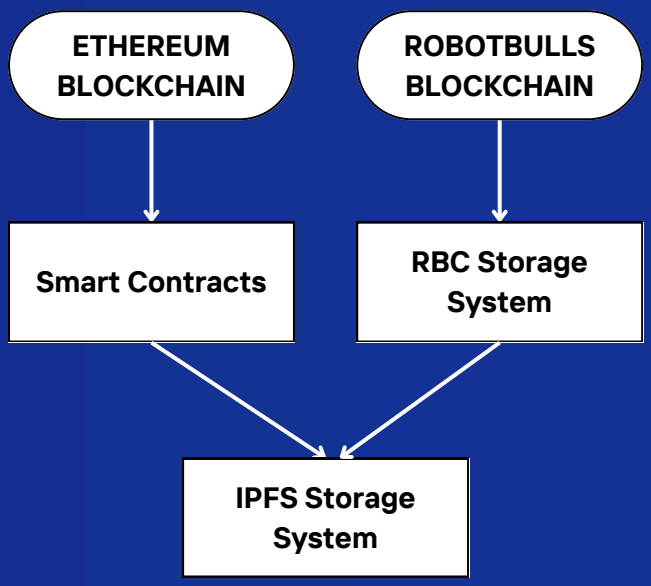
By combining blockchain execution, decentralized storage, and permissionless governance, RobotBulls provides a resilient, transparent, and adaptable infrastructure for decentralized algorithmic trading.

1. Introduction

The advent of blockchain technology has catalyzed the emergence of decentralized finance (DeFi) platforms, fundamentally altering conventional financial paradigms. RobotBulls embodies a pioneering approach to trading platforms by integrating blockchain capabilities with open-source innovation to establish a permissionless framework where trading robots can be developed, reviewed, and executed entirely on-chain. This section elucidates the fundamental concepts and objectives underpinning the RobotBulls platform, providing a comprehensive overview of its mission and vision.

2. Platform Architecture

RobotBulls integrates blockchain execution, decentralized storage, and governance into a unified protocol design.



2.1 Smart Contracts

Smart contracts govern robot registration, execution, staking, and governance. Written in Solidity, they are deployed on Ethereum and the RobotBulls blockchain. These contracts ensure deterministic, transparent execution and cannot be altered once deployed.

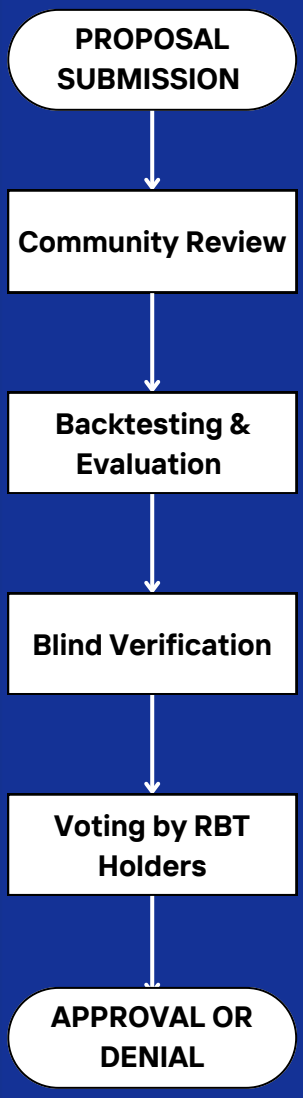
2.2 Decentralized Storage

Robot code and associated metadata are stored using the InterPlanetary File System (IPFS). Each robot is identified by a unique content hash, ensuring verifiability, immutability, and censorship resistance.

2.3 Interoperability

The protocol is fully compatible with the Ethereum Virtual Machine (EVM), enabling deployment on Ethereum mainnet and Layer 2 scaling solutions such as Optimistic Rollups and zk-Rollups.

Cross-chain expansion is supported through adapters, allowing RobotBulls to operate across heterogeneous blockchain environments. This design maximizes reach and flexibility while preserving security and transparency.



3. Functionality

3.1 Proposal Submission

Developers publish a robot by submitting metadata and code references to the registry contract. This process is permissionless, ensuring global accessibility.

3.2 Community Review

Community participants evaluate submissions on dimensions such as transparency, robustness, and adherence to secure coding practices.

3.3 Backtesting and Verification

Robots may undergo backtesting using historical datasets to simulate performance across market conditions, following best practices in financial research. Blind verification enables evaluation without disclosing proprietary logic, ensuring impartial assessment.

3.4 Deployment

Robots that pass review are made available for deployment. Users configure individual parameters and authorize smart contracts to execute directly from their wallets. All activity is conducted in a non-custodial manner, with each user retaining direct control over their assets.

4. Utilizing Trading Robots

The interaction flow for users is as follows:

- **Selection**: Review available robots, including metadata and open performance indicators.
- **Configuration**: Define parameters such as execution duration and allocation.
- **Execution**: Authorize a smart contract to run trades from the user's wallet.
- **Redemption**: Withdraw or reconfigure assets at any point.

This flow ensures independent execution while maintaining user custody of funds.

5. Governance

5.1 DAO Framework

RobotBulls governance is managed by a Decentralized Autonomous Organization (DAO). Its responsibilities include protocol upgrades, parameter tuning, and security-related adjustments. Governance focuses exclusively on technical and systemic parameters, not individual robot approval.

5.2 Voting System

The RobotBulls Token (RBT) is used for governance. Voting rights are proportional to token holdings, with quadratic mechanisms considered for future iterations to encourage balanced participation.

6. Tokenomics

6.1 RBT – Governance Token

- The sole token of the protocol, RBT, is designed for governance only.
- It enables submission and voting on proposals.
- It holds no financial or ownership rights.
- Its purpose is limited to decentralized coordination.

7. Security and Decentralization

7.1 Smart Contract Security

RobotBulls contracts are open-source and undergo community and external audits. Once stable, governance processes ensure immutability of deployed contracts.

7.2 User Custody

Users maintain direct control of their private keys and funds at all times. No pooled custody exists at the protocol level.

7.3 Forkability

RobotBulls is released under the MIT license, ensuring that anyone may fork or adapt the protocol. This guarantees resilience, innovation, and independence of the community.

8. Conclusion

RobotBulls introduces a decentralized framework for algorithmic trading. Through smart contracts, decentralized storage, and community governance, it provides an open, transparent, and censorship-resistant infrastructure for deploying trading robots.
Its permissionless architecture empowers developers, users, and communities to collaborate in advancing decentralized trading technology, while its open-source nature ensures adaptability and resilience across blockchain ecosystems.

References

1. Benet, J. (2014). IPFS - Content Addressed, Versioned, P2P File System (DRAFT 3). https://github.com/ipfs/papers/raw/master/ipfs-cap2pfs/ipfs-p2p-file-system.pdf
2. Baer, K., Campollo, A., & Hardie, A. (2018). Backtesting and Its Pitfalls. Journal of Portfolio Management, 44(6), 81-91. https://doi.org/10.3905/jpm.2018.44.6.081
3. Buterin, V. (2013). A Next-Generation Smart Contract and Decentralized Application Platform. https://ethereum.org/en/whitepaper/