White Paper

Storage Protocol

Web 3.0

Storage

Blockchain

Payments

Network ID

420500100

RPC

rpc.storageprotocol.com

Table of Contents

1. Introduction

The Storage Protocol whitepaper introduces a revolutionary decentralized storage solution powered by blockchain technology. This section outlines the primary objectives and introduces CubitsSDK, a pivotal component enhancing developer capabilities within the StorageProtocol ecosystem.

2. Vision and Objectives

Storage Protocol aims to address key challenges in decentralized storage, such as user control, complexity, and cost. The objectives include providing a user-friendly, cost-effective, and secure storage solution for businesses and individuals. CubitsSDK aligns with this vision by empowering developers to build innovative decentralized applications seamlessly.

3. Technical Overview

3.1 Network Architecture

The Storage Protocol operates on an EVM-based blockchain with IPFS integration for storage. It employs a Proof of Authority (PoA) consensus mechanism to ensure efficiency and security. CubitsSDK integration enhances the technical landscape by providing developers with a comprehensive toolkit for blockchain interaction.

3.2 Network Architecture

The PoA mechanism involves elected council members who run transaction pools, ensuring network continuity and integrity. CubitsSDK does not directly impact the consensus mechanism but augments its functionality by enabling developers to interact with multisig contracts and access network events.

3.3 Storage Protocol Operation

Nodes earn rewards in Cubits, the network's native currency, for providing storage. The protocol regulates node rewards and storage costs through smart contracts. CubitsSDK integrates seamlessly with the Storage Protocol's operation, offering developers access to essential functions and event hooks for building decentralized applications.

3.4 Storage Protocol Operation

CubitsSDK empowers developers with a comprehensive set of function APIs, event hooks, and utility functions to interact effectively with the blockchain network. The SDK facilitates seamless integration with the Storage Protocol, enabling developers to build decentralized applications with ease.

4. Cubits: The Network Currency

Cubits serve as the primary currency within the Storage Protocol network, facilitating transactions and rewards. CubitsSDK facilitates transactions involving Cubits by providing APIs for managing Cubits transactions and accessing related data.

5. Smart Contracts and Governance

5.1 Contract Functions

Smart contracts govern various aspects of the network, including storage contracts, USDT integration, node management, and multisig operations. CubitsSDK empowers developers to interact with these smart contracts seamlessly.

5.2 Contract Functions

Storage Protocol employs a governance model where active members participate in decision-making based on Cubit holdings and consensus history. CubitsSDK allows developers to participate in the governance of the network by accessing relevant contract functions and monitoring network events.

6. Use Cases

6.1 Business Applications

Storage Protocol enables businesses to store essential data securely on the blockchain for applications such as delivery jobs, insurance claims, and government filings. CubitsSDK empowers

developers to build decentralized applications tailored to specific business needs.

6.2 Private Use

Individuals can use Storage Protocol for personal data storage, ensuring privacy and security through advanced encryption technologies. CubitsSDK provides developers with tools to develop personal applications for storing sensitive data securely on the blockchain and managing financial transactions with ease.

7. Network Operation

7.1 Node Requirements

Nodes require specific hardware and storage capacity to participate in the network and earn rewards. CubitsSDK does not directly impact node requirements but provides developers with tools to interact with node-related data and events.

7.2 Block Capacity and Gas Limit

The network's block capacity and gas limit ensure efficient transaction processing and resource utilization. Developers can utilize CubitsSDK to monitor block capacity and gas limits, ensuring smooth operation within the Storage Protocol network.

8. Stable Coin Integration

8.1 Stable Coin Integration

Storage Protocol integrates a stablecoin, SRC-20 USDT Token, for seamless transactions and stability. CubitsSDK supports stablecoin integration by providing APIs for managing USDT transactions and accessing stablecoin-related data.

8.2 Stable Coin Integration

Multisig contracts and regular code audits ensure the security and integrity of stablecoin transactions within the Storage Protocol network. Developers can leverage CubitsSDK to interact with multisig

contracts and implement additional security measures within their decentralized applications.

9. Community and Development

9.1 Community Engagement

The project fosters community involvement through open-source development and governance participation. CubitsSDK empowers developers to contribute to the Storage Protocol ecosystem by providing tools and resources for building decentralized applications and enhancing network functionality.

9.2 Open Source Development

Storage Protocol encourages open-source contributions to enhance security, transparency, and innovation. Developers are encouraged to contribute to CubitsSDK's development, ensuring its continued growth and success within the Storage Protocol ecosystem.

10. Additional Ideas and Future Directions

Storage Protocol is not limited to existing use cases and can serve as a backbone for various technological innovations. CubitsSDK serves as a catalyst for innovation within the Storage Protocol ecosystem, enabling developers to explore new ideas and build cutting-edge decentralized applications.

11. Partnerships and Marketing

The project collaborates with partners to expand its reach and promote adoption. Marketing efforts focus on technology innovation and decentralized principles. Strategic partnerships aim to showcase the capabilities of CubitsSDK and attract developers to the Storage Protocol ecosystem.

12. Conclusion

The Storage Protocol whitepaper outlines a robust decentralized storage solution with a focus on user experience, security, and innovation. The protocol offers a strong foundation for businesses and individuals seeking secure and cost-effective storage solutions in the web 3.0 era. CubitsSDK serves as a powerful tool

for developers building on the Storage Protocol blockchain, empowering them with a rich set of function APIs, event hooks, and utility functions to create innovative decentralized applications seamlessly. By providing seamless interaction with the blockchain network, CubitsSDK contributes to the growth and adoption of the Storage Protocol ecosystem.

13. Appendix

13.1 Contracts

CUBITS CONTRACT

0x1111111111111111111111111111111111111111

This contract allows wallets holding Cubits to create storage contracts, enabling contract owners full ownership and control over the contract. Storage costs are calculated based on factors such as storage length, replication factor, and size.

USDT CONTRACT

0x7777777777777777777777777777777777777777

The USDT contract facilitates the exchange of USDT tokens within the StorageProtocol network, enabling seamless transactions and stability. It also tracks Cubits in circulation and sets the token ratio.

NODE CONTRACT

0x2222222222222222222222222222222222222222

The Node contract records all nodes in the network and handles their payouts. It adjusts collateral requirements based on network conditions, ensuring stability and efficiency.

MULTISIG CONTRACT

0x9999999999999999999999999999999999999999

The Multisig contract coordinates cross-chain transactions and serves as a custodian for development rewards. Multisig members verify transactions and execute decisions on the network.

MULTISIG CONTROLLER CONTRACT

0x9990000000000000000000000000000123456789

The Multisig Controller contract manages network actions and settings, such as minting/burning USDT and adjusting network parameters.

13.2 Important Links

StorageProtocol Website: www.storageprotocol.com
Explore for updates and information.

Development Team Contact: team@storageprotocol.com
Connect with our team via email.

Blockchain Explorer: https://explorer.storageprotocol.com
Dive into network exploration.

Technical Documentation: https://info.storageprotocol.com
Access detailed project resources.

Node Dashboard: https://nodes.storageprotocol.com
Monitor network nodes effortlessly.

Source Code Repository: https://github.com/storageprotocol
Collaborate on our codebase.

Web Wallet Access: https://wallet.storageprotocol.com
Manage assets conveniently.

Network Statistics: https://stats.storageprotocol.com
Stay informed with real-time statistics.

Governance Platform: https://vote.storageprotocol.com
Shape the network's future.

Network Information Hub: https://info.storageprotocol.com
Access detailed network insights.

Web3 API Endpoint: https://api.storageprotocol.com
Integrate seamlessly with our API.

13.3 Nodes

Collateral:

Required and set by the Multisig according to economic evaluation. Existing nodes remain valid even if collateral increases or decreases.

Node Requirements:

500GB+ storage capacity

80 GB of available storage

4 GB of RAM

Public static IP address

Block Capacity:

100 Million Gas