Introducing ERC-884: Ethereum's Membership Token Standard

Nova Novriansyah · Follow
Published in Novai-Blockchain 101
5 min read · May 7, 2024

Listen Share

Blockchain technology has revolutionized various aspects of our lives, from finance to supply chain management. Ethereum, with its smart contract capabilities, enables the creation of various tokens representing different assets and utilities. One such token standard is ERC-884, designed specifically for managing membership rights and privileges on the Ethereum blockchain.

What is ERC-884?

ERC-884 is an Ethereum token standard that defines the functionalities required for managing membership-based systems on the blockchain. It allows developers to create tokens that represent memberships in organizations, clubs, communities, or any other group where membership privileges are granted.

Features of ERC-884:

1. Membership Management: ERC-884 tokens enable the management of memberships, including membership issuance, renewal, revocation, and transferability.

2. Privilege Assignment: The standard allows for the assignment of specific privileges or access rights to members based on their token holdings.

3. Dynamic Membership Levels: Developers can implement dynamic membership levels or tiers, where different levels of membership come with varying

privileges or benefits.

4. Member Voting Rights: ERC-884 tokens can incorporate voting mechanisms, allowing members to participate in governance decisions within the organization.
5. Interoperability: ERC-884 tokens can interact with other Ethereum-based protocols and decentralized applications (dApps), enhancing their utility and integration capabilities.

Real Case for EFC 884

The ERC-884 standard, designed for managing membership rights and privileges on the Ethereum blockchain, can be applied to various real-life scenarios where membership-based systems are prevalent. Here are some examples:

1. Clubs and Associations: Traditional clubs and associations, such as sports clubs, hobby groups, or professional organizations, can implement ERC-884 tokens to represent membership. These tokens can grant access to club facilities, events, or exclusive resources, and they can also serve as a means of voting for club decisions or electing leaders.
2. Subscription Services: Online subscription-based platforms, such as streaming services, educational platforms, or software subscription services, can utilize ERC-884 tokens to manage user memberships. Token holders can enjoy subscription benefits, access premium content, or participate in platform governance through voting mechanisms.
3. Gyms and Fitness Centers: Fitness centers and gyms can issue ERC-884 tokens to their members, granting access to gym facilities, classes, or personal training sessions. Members can use their tokens to book appointments, track attendance, and even earn rewards based on their level of engagement.
4. Coworking Spaces: Coworking spaces and shared offices can adopt ERC-884 tokens as a means of managing memberships and access privileges. Token holders can use their memberships to reserve desk space, access amenities like conference rooms or printing facilities, and participate in community events or networking opportunities.

5. Professional Networks: Professional networks or industry associations can implement ERC-884 tokens to represent membership in their communities. Token holders may gain access to job postings, mentorship programs, exclusive networking events, or industry reports. Additionally, tokens can serve as a reputation mechanism within the network, reflecting members’ contributions and expertise.

6. Membership Rewards Programs: Retailers or service providers can create ERC-884 tokens to reward customer loyalty and incentivize repeat business. Members can earn tokens based on their purchases or engagement with the brand and redeem them for discounts, exclusive offers, or branded merchandise.

Sample ERC-884 Contract:

Below is a simplified example of an ERC-884 contract written in Solidity:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.0;

/**
 * @title ERC-884 Membership Token Standard
 * @dev Interface for a Membership Token contract, which represents membership
 */
interface IERC884 {
    /**
     * @dev Emitted when `member` is granted `membershipId` membership.
     */
    event MembershipGranted(address indexed member, uint256 indexed membershipI

    /**
     * @dev Emitted when `membershipId` membership is revoked from `member`.
     */
    event MembershipRevoked(address indexed member, uint256 indexed membershipI

    /**
     * @dev Returns the total supply of membership tokens.
     */
    function totalSupply() external view returns (uint256);

    /**
     * @dev Returns the balance of the `member` address.
     */
    function balanceOf(address member) external view returns (uint256);

    /**
     * @dev Grants membership to `member` for the given `membershipId`.
     */
    function grantMembership(address member, uint256 membershipId) external ret

    /**
     * @dev Revokes membership from `member` for the given `membershipId`.
     */
    function revokeMembership(address member, uint256 membershipId) external re

    /**
     * @dev Checks if `member` holds the given `membershipId`.
     */
    function hasMembership(address member, uint256 membershipId) external view
}

/**
 * @title ERC-884 Membership Token Contract
 * @dev Implementation of the ERC-884 Membership Token Standard.
 */
contract ERC884 is IERC884 {
    // Mapping from member address to their memberships
    mapping(address => mapping(uint256 => bool)) private _memberships;

    // Total supply of membership tokens
    uint256 private _totalSupply;

    /**
     * @dev Grants membership to `member` for the given `membershipId`.
     * Emits a {MembershipGranted} event.
     */
    function grantMembership(address member, uint256 membershipId) external ove
        require(!_memberships[member][membershipId], "Membership already grante
        _memberships[member][membershipId] = true;
        _totalSupply++;
        emit MembershipGranted(member, membershipId);
        return true;
    }

    /**
     * @dev Revokes membership from `member` for the given `membershipId`.
     * Emits a {MembershipRevoked} event.
     */
    function revokeMembership(address member, uint256 membershipId) external ov
        require(_memberships[member][membershipId], "Membership not granted");
        _memberships[member][membershipId] = false;
        _totalSupply--;
        emit MembershipRevoked(member, membershipId);
        return true;
    }

    /**
     * @dev Returns the total supply of membership tokens.
     */
    function totalSupply() external view override returns (uint256) {
        return _totalSupply;
    }

    /**
     * @dev Returns the balance of the `member` address.
     */
    function balanceOf(address member) external view override returns (uint256)
        uint256 count;
        for (uint256 i = 0; i < _totalSupply; i++) {
            if (_memberships[member][i]) {
                count++;
            }
        }
        return count;
    }

    /**
     * @dev Checks if `member` holds the given `membershipId`.
     */
    function hasMembership(address member, uint256 membershipId) external view
        return _memberships[member][membershipId];
    }
}
```

This contract implements the ERC-884 Membership Token Standard, providing functions to grant and revoke membership rights, check membership balances, and verify membership status. It emits events when memberships are granted or revoked.

Deployment Instructions:

To deploy the ERC-884 contract using Truffle, follow these steps:

1. Install Truffle globally using npm if you haven't already: `npm install -g truffle`.

2. Create a new directory for your project and navigate into it.
3. Initialize a new Truffle project: `truffle init`.
4. Replace the contents of the `contracts/MembershipToken.sol` file with the Solidity code provided above.
5. Compile the contract: `truffle compile`.
6. Configure your deployment settings in the `truffle-config.js` file, including the Ethereum network you want to deploy to.
7. Run the migration to deploy the contract: `truffle migrate --network <network_name>`, replacing `<network_name>` with the name of the desired network (e.g., `mainnet`, `rinkeby`, `ropsten`, etc.).

After successful deployment, you'll receive the contract address, which you can use to interact with the ERC-884 tokens on the specified Ethereum network.

Erc884 Ethereum Blockchain Web3

Follow

Published in Novai-Blockchain 101

1 Follower · Last published Jun 2, 2024

Welcome to our blockchain channel, where we unravel the mysteries of decentralized technology. Delve into the concepts of public and private blockchains, exploring their unique features, applications, and potential impact on various industries. Whether you're a blockchain novice

Follow

Written by Nova Novriansyah

109 Followers · 34 Following

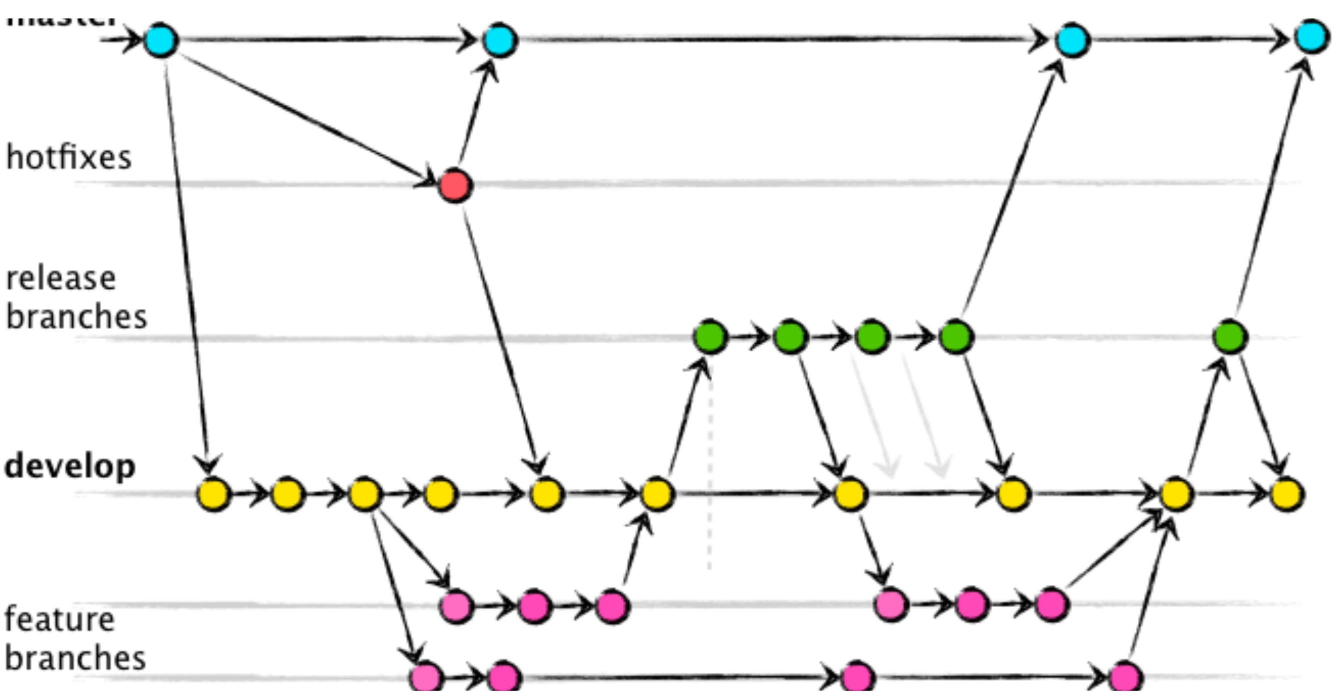
C|CISO, CEH, CC, CVA,CertBlockchainPractitioner, Google Machine Learning , Tensorflow, Unity Cert, Arduino Cert, AWS Arch Cert. CTO, IT leaders. Platform owners

No responses yet

What are your thoughts?

Respond

More from Nova Novriansyah and Novai-Blockchain 101

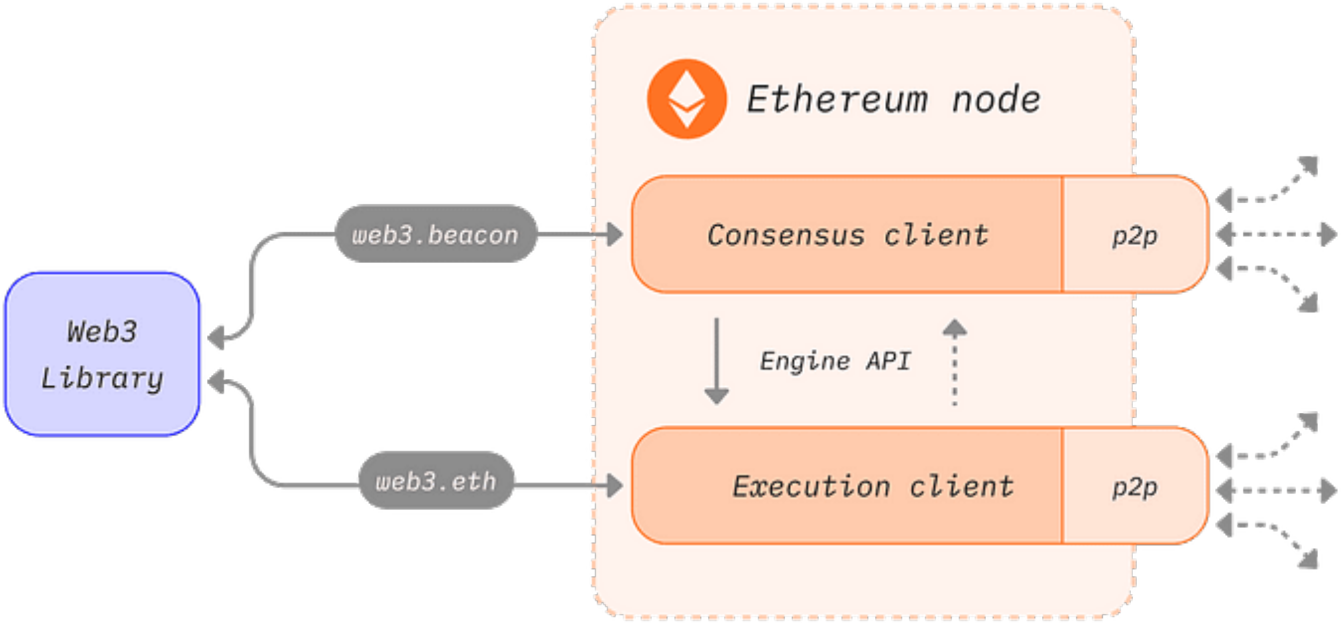


In NovAI- Agile & DevOPS 101 by Nova Novriansyah

Top 4 Branching Strategies and Their Comparison: A Guide with Recommendations

Branching strategies are critical in version control, helping teams manage and organize code changes efficiently. Choosing the right...

Aug 15 👏 14

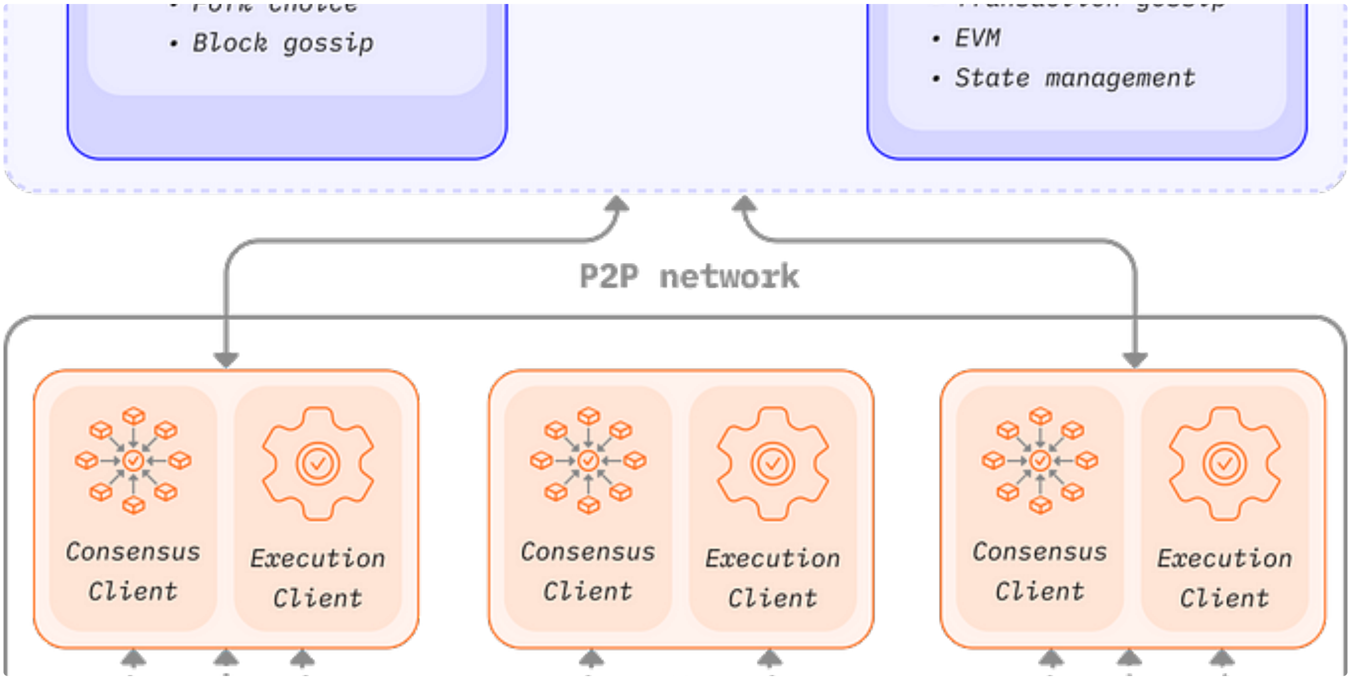


In Novai-Blockchain 101 by Nova Novriansyah

Understanding Nodes and Clients in Ethereum

In the realm of Ethereum, nodes and clients play crucial roles in maintaining the network's integrity and facilitating transactions. Let's...

May 7 👏 2



In Novai-Blockchain 101 by Nova Novriansyah

Understanding Ethereum Node Architecture

Ethereum, the groundbreaking blockchain platform, operates through a complex network of nodes. These nodes play crucial roles in executing...

May 7 👏 2

In NovAI Cloud Computing — GCP by Nova Novriansyah

How to Install Google Cloud CLI (Command-Line Interface) on Mac, Windows, and Linux

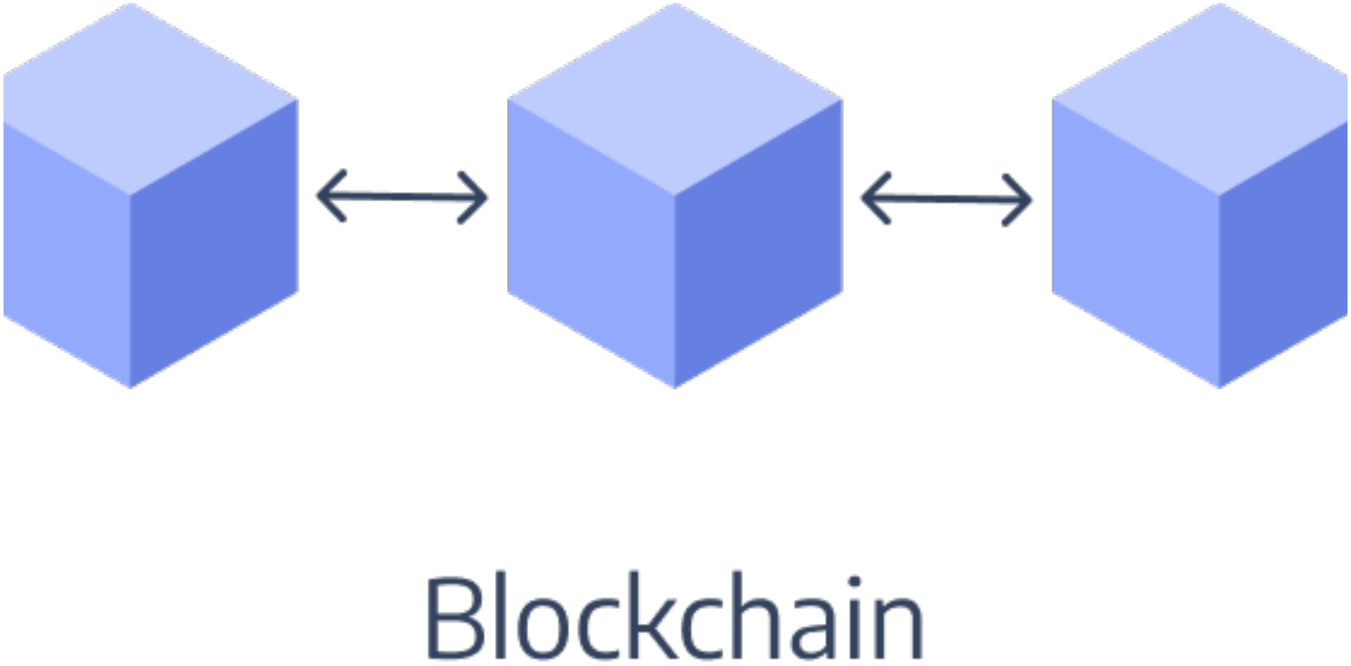
Google Cloud CLI, known as gcloud, is an essential tool for managing Google Cloud Platform (GCP) resources from the command line...

Jun 21 1 1

See all from Nova Novriansyah

See all from Novai-Blockchain 101

Recommended from Medium



Sithara Wanigasooriya

Blockchain in 2024: An Expert's Guide to its Core Components and Evolution

Blockchain is now recognized as a decentralized, secure, and transparent way to store and manage data across a network of computers without...

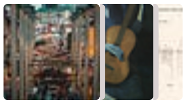
AK

Web3 and Blockchain Development in 2024: A Comprehensive Engineering Guide

After leading blockchain development teams at major financial institutions and implementing numerous Web3 solutions, I've learned that...

Nov 25 10

Lists

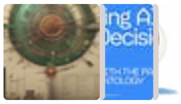


My Kind Of Medium (All-Time Faves)

102 stories · 598 saves

MODERN MARKETING

199 stories · 948 saves

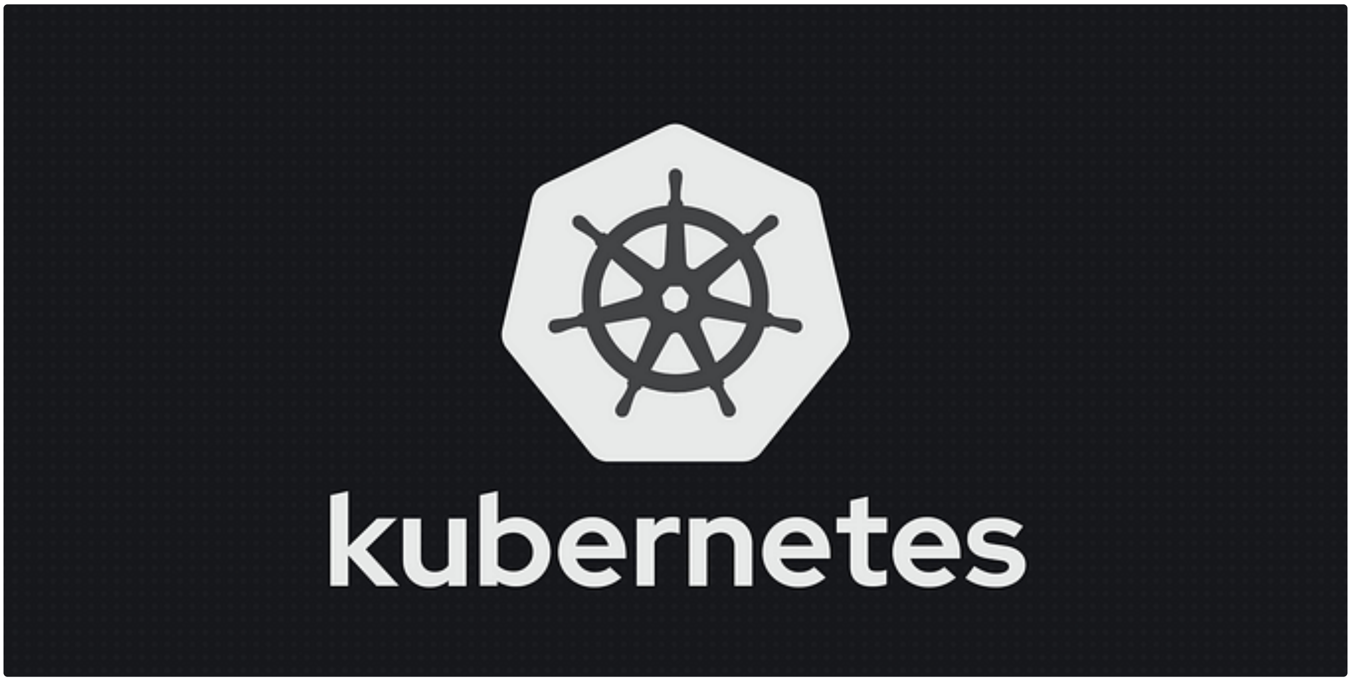


data science and AI

40 stories · 296 saves

Generative AI Recommended Reading

52 stories · 1532 saves



In Stackademic by Crafting-Code

I Stopped Using Kubernetes. Our DevOps Team Is Happier Than Ever

Why Letting Go of Kubernetes Worked for Us

✦ Nov 19 3.7K 111

Lovish Kumar

Encrypt and Decrypt JWT Token using RSA Algorithm in Node.js

In modern applications, JWT (JSON Web Tokens) are widely used for authentication and authorization. Typically, JWTs are signed but not...

Sep 4 11

Tushar Bhatia

Exploring Solana's Consensus- What Makes Proof of Stake and Proof of History So Fast?

If you're curious about what makes Solana stand out in the blockchain world, you're not alone. Honestly, when I first dug into it, I was...

Nov 1 50

Ulltimate guide to selecting tp
cryptoccurreency
walllet for safee transmations

Prashanth Noble Bose

Ultimate Guide to Selecting the Top Cryptocurrency Wallet for Safe Transactions

Ultimate Guide to Selecting the Top Cryptocurrency Wallet for Safe Transactions

1d ago

See more recommendations