NEAR Accounts

Users participate in the NEAR ecosystem through their NEAR accounts. These accounts are identified by a unique address , can optionally hold a smart contract , and are controlled through Access Keys .

By signing transactions with their account, users can:

1. Send and receive digital assets
2. (such as tokens or collectibles)
3. Create and interact with on-chain applications known as smart contracts
4. Control accounts in other chains
5. (such as Ethereum or Bitcoin) ✧
6. Help onboard new users by covering the costs
7. of their transactions (gas fees)

Want to create an account? You have multiple ways to create an account, you can sign-up using your email , get a mobile wallet through telegram , or create a web wallet .

Account Model Overview

Let's take a closer look at the different elements that compose the NEAR account model.

Account ID

NEAR implements two types of accounts IDs: named accounts such as alice.near , which are simple to remember and share, and the classic alphanumeric IDs (fb9243ce...) that other chains also implement.

Permissions Through Access Keys

NEAR accounts can have multiple keys , each with their own set of permissions. This allows to easily swap keys if one gets compromised, and to use keys as authorization tokens for third-parties.

Simple to Develop Smart Contracts

NEAR accounts can optionally hold a simple program, known as a smart contract . In NEAR, developers can create smart contracts using languages such as Javascript or Rust.

Comparison With Ethereum

If you're familiar with development on Ethereum, it's worth making a quick note about how accounts are different. The table below summarizes some key differences:

Ethereum Wallet NEAR Account Public Identifier Public Key (0x123...) Named IDs (alice.near) and implicit accounts (Public Key 0x123...) Secret Key Private Key (0x456...) Multiple key-pairs with permissions: -FullAccess key -FunctionCall key Characteristics - Private key gives full access - Account doesn't have to be created via a transaction - Permission-based keypair - Account ID must be created via blockchain transaction Edit this page Last updated on Mar 25, 2024 by gagdiez Was this page helpful? Yes No