BCDV 1010
Smart Contract Development Essentials

2023 January

week 02 - class 09

Tokens

What is a Token?

Tokens are smart contracts which can represent virtually anything in Ethereum:

- reputation points in an online platform
- skills of a character in a game
- lottery tickets
- financial assets like a share in a company
- a fiat currency like USD
- an ounce of gold
- and more...

Fungible vs Non-Fungible Tokens

Fungible Tokens	Non-Fungible Tokens
1. They are interchangeable	1. They are unique in nature and can not be interchanged.
2. They are a store of value	2. They are a store of data
3. The tokens are built mostly on ERC-20 standards	3. The NFTs are ERC-721 standard tokens
4. Divisible into smaller parts	4. NFTs are indivisible and only have value as a whole.
5. E.g: Cryptocurrencies	5. E.g: CryptoPunks

Token Standards

Many Ethereum development standards focus on token interfaces. These standards help ensure smart contracts remain composable, so when a new project issues a token, it remains compatible with existing decentralized exchanges

Some of the most popular token standards on Ethereum:

- ERC-20 - A standard interface for fungible (interchangeable) tokens, like voting tokens, staking tokens or virtual currencies.
- ERC-721 - A standard interface for non-fungible tokens, like a deed for artwork or a song.
- ERC-777 - ERC-777 allows people to build extra functionality on top of tokens such as a mixer contract for improved transaction privacy or an emergency recover function to bail you out if you lose your private keys.
- ERC-1155 - ERC-1155 allows for more efficient trades and bundling of transactions – thus saving costs. This token standard allows for creating both utility tokens (such as $BNB or $BAT) and Non-Fungible Tokens like CryptoPunks.
- ERC-4626 - A tokenized vault standard designed to optimize and unify the technical parameters of yield-bearing vaults.

ERC-20

The ERC-20 (Ethereum Request for Comments 20), proposed by Fabian Vogelsteller in November 2015, is a Token Standard that implements an API for tokens within Smart Contracts.

Example functionalities ERC-20 provides:

- Transfer tokens from one account to another
- Get the current token balance of an account
- Get the total supply of the token available on the network
- Approve whether an amount of token from an account can be spent by a third-party account

If a Smart Contract implements the following methods and events it can be called an ERC-20 Token Contract and, once deployed, it will be responsible to keep track of the created tokens on Ethereum.

Methods:

```
function name() public view returns (string)
function symbol() public view returns (string)
function decimals() public view returns (uint8)
function totalSupply() public view returns (uint256)
function balanceOf(address _owner) public view returns (uint256 balance)
function transfer(address _to, uint256 _value) public returns (bool success)
function transferFrom(address _from, address _to, uint256 _value) public returns (bool s
function approve(address _spender, uint256 _value) public returns (bool success)
function allowance(address _owner, address _spender) public view returns (uint256 remain
```

Events:

```
event Transfer(address indexed _from, address indexed _to, uint256 _value)
event Approval(address indexed _owner, address indexed _spender, uint256 _value)
```

OpenZepplin

A library of modular, reusable, secure smart contracts for the Ethereum network, written in Solidity. OpenZeppelin provides OpenZepplin Open-source Smart contracts that provide the secure implementation of the Ethereum Standards such as ERC-20, ERC-721 etc.

The OpenZepplin smart contracts can be accessed on GitHub:
https://github.com/openzeppelin

OpenZepplin is a new standard in the industry while writing smart contracts.

Token Extensions

We can include the following functionalities in a token to manage the supply of the tokens

minting:
minting is the creation of new tokens on the blockchain through computational processes to validate information, create new blocks, and record information on the blockchain.

burning:

burning is the process by which users can remove tokens (also called coins) from circulation, which reduces the number of coins in use. The tokens are sent to a wallet address that cannot be used for transactions other than receiving the coins. The wallet is outside the network, and the tokens can no longer be used.

pausing:

A token pause transaction prevents the token from being involved in any kind of operation. The token's pause key is required to sign the transaction. This is a key that is specified during the creation of a token. If a token has no pause key, you will not be able to pause the token.