"So you can't own the precious physically, but you can pay to have your name listed as its owner in an online distributed database."

CS 168: Blockchain and Cryptocurrencies

Oracles and Tokens

Prof. Tom Austin
San José State University

Oracles

Motivation

- EVM execution must be deterministic.
  - Cannot rely on outside information
- But sometimes that information is important:
  - Supply chain tracking
  - Exchange rate data
  - Weather data

Solution: Oracle

Oracle writes transaction to blockchain

- Must be trusted party, or group.
- Transaction includes additional data.
- Signed messages.

Three Oracle Designs

- Immediate-read
  - Data stored in contract
  - E.g. academic certificates, club membership, etc.
- Publish-subscribe
  - Used for frequently changing data
    - E.g. stock prices, weather, etc.
  - Off-chain daemons watch for updates on-chain
- Request-response
  - Data too large to store on blockchain
  - Co-ordinates with off-chain system on demand

Computation Oracles

- Trusted third party that performs computation off-chain.
- Used for efficiency reasons.

Lab, part 1: Temperature Oracle

We will write a simple oracle for recording temperatures on the blockchain, storing the temperatures by zip code.

Details in Canvas.

Tokens

What is a token?

Represent some resource or rights:

- Access rights
- Placeholder for real-world asset
- Alternate currency
  - Frequently used for Initial Coin Offerings (ICOs).

Ethereum Tokens

Minimal viable token must have:

- mapping of accounts to balances
- transfer function

See https://www.ethereum.org/token has example, copied on next slide.

```
contract MyToken {
  mapping (address => uint256) public balanceOf;

  constructor(uint256 initialSupply) {
    balanceOf[msg.sender] = initialSupply;
  }

  function transfer(address to,
        uint256 value) public returns (bool) {

    require(balanceOf[msg.sender] >= value);
    require(balanceOf[to] + value >= balanceOf[to]);

    balanceOf[msg.sender] -= value;
    balanceOf[to] += value;

    return true;
  }
}
```

ERC-20 Tokens

- Ethereum Request for Comment (ERC)
  - Proposed by Fabian Vogelsteller
  - Assigned issue #20 by Github automatically
- Became Ethereum Improvement Proposal 20 (EIP-20), but ERC-20 name stuck.
- Defines common interface for fungible tokens.

Required ERC-20 Functions

```
totalSupply()
balanceOf(address tokenOwner)
transfer(address to, uint tokens)

approve(address spender, uint tokens)
allowance(address tokenOwner,
    address spender)
transferFrom(address from,
    address to, uint tokens)
```

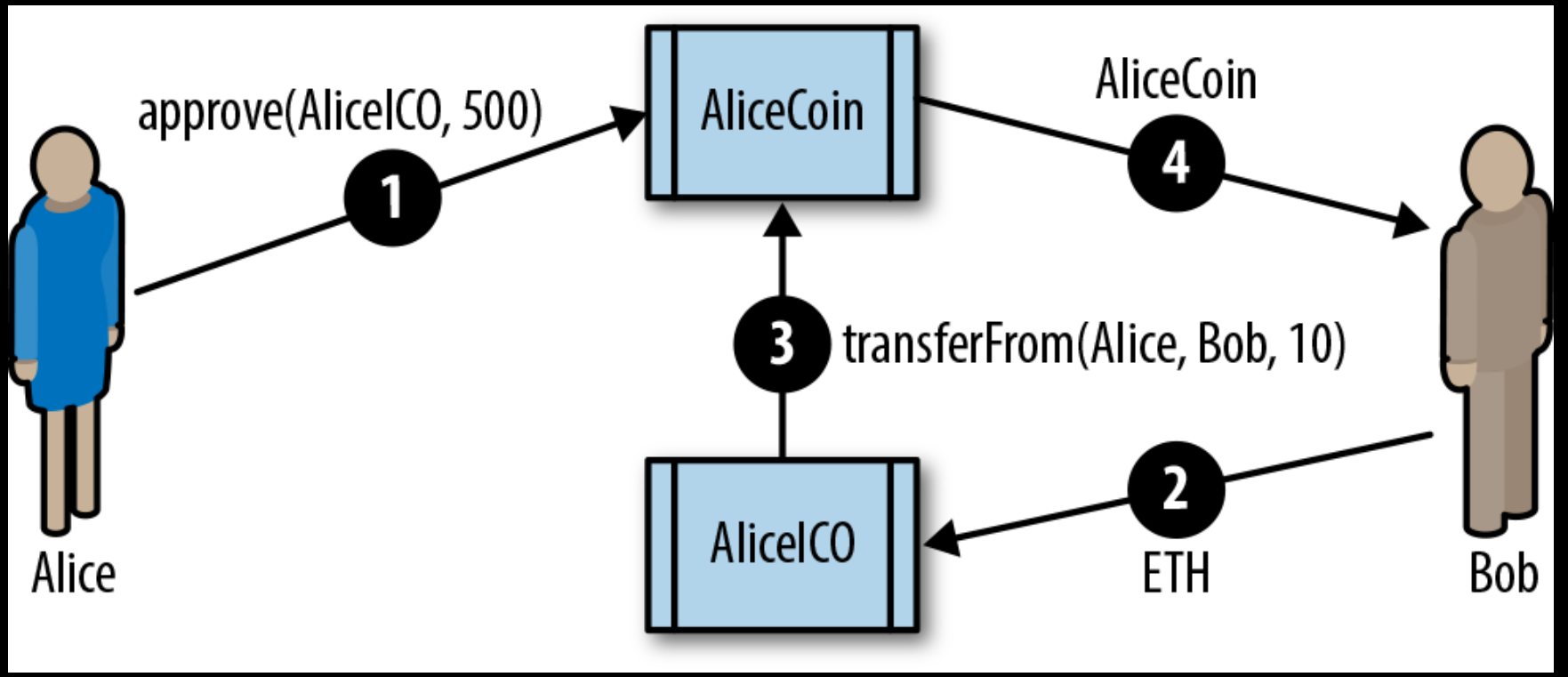
`approve`, `allowance`, and `transferFrom`

- Allow another user to withdraw your tokens
  - Smart contracts
  - Exchanges
- `approve` grants access to funds
- `allowance` shows the amount available
- `transferFrom` transfers funds to another account

`approve`, `allowance`, and `transferFrom` process

1. Alice uses `approve` to grant AliceICO smart contract 500 tokens
2. Bob calls AliceICO to buy tokens with ether
3. AliceICO uses `transferFrom` to give Alice's tokens to Bob
4. Tokens are transferred to Bob

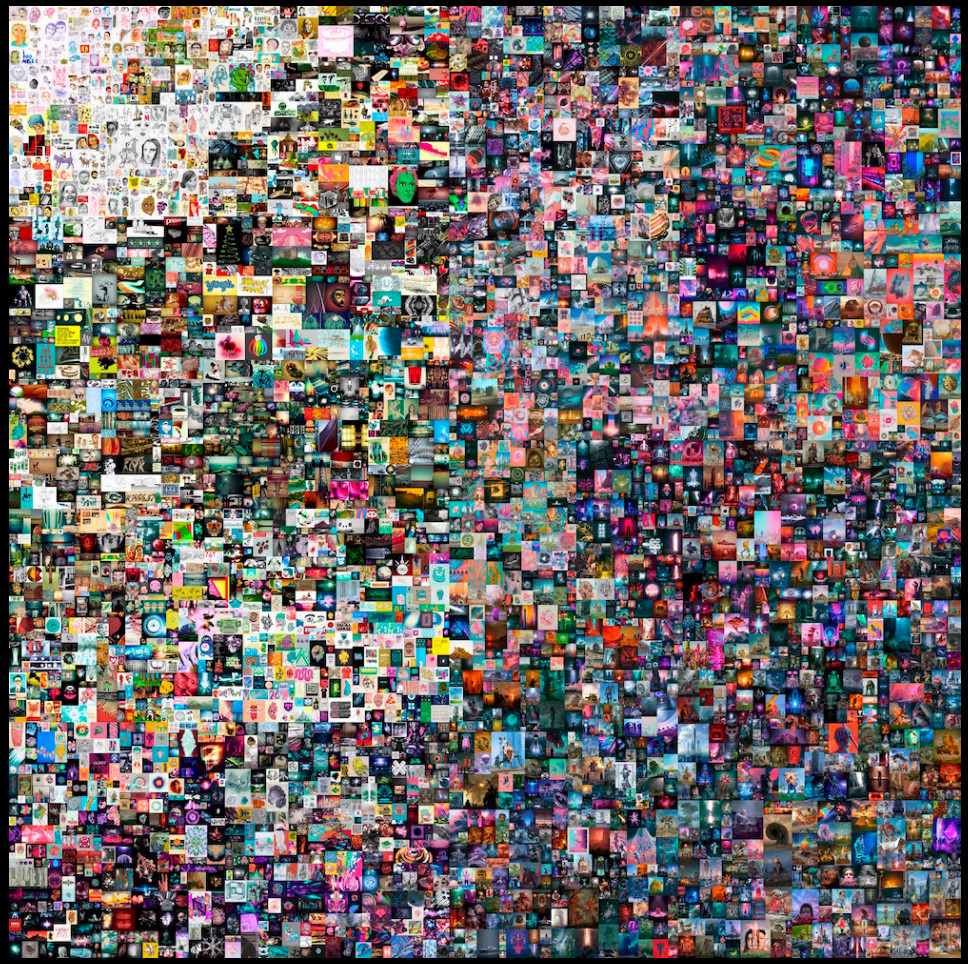
`approve`, `allowance`, and `transferFrom` illustrated (courtesy of *Mastering Ethereum*)



ERC-20 Optional Functions

- **`name`** – human-readable name of the token.
- **`symbol`** – human-readable token symbol.
- **`decimals`** – # decimals used to divide token amounts.

Beeple's "*Everydays: the First 5000 Days*"



- Collage of 5000 digital images
- NFT auctioned at Christie's in 2021
- Sold for 42,329 ETH
  - $69.3 million

Non-Fungible Tokens (NFTs)

- Unique digital asset
- *Might* represent a real-world asset
- Use cases:
  - Games
  - Inventory management
  - Digital artwork

NFT History

- 2014 – First NFT launched on Namecoin
- 2017:
  - CryptoPunks app – 1st ETH NFT
  - CryptoKitties released
- 2018:
  - ERC-165 standard for interface detection
  - ERC-721 standard for NFTS proposed

Required ERC-721 Functions

```
balanceOf(address owner)
ownerOf(uint256 tokenId)

approve(address addr, uint256 tokenId)
getApproved(uint256 tokenId)
setApprovalForAll(address operator, bool approved)
isApprovedForAll(address owner, address operator)

transferFrom(address from, address to, uint256 tokenId)
safeTransferFrom(address from, address to,
      uint256 tokenId)
safeTransferFrom(address from, address to,
      uint256 tokenId, bytes data)
```

Lab, part 2: A Token of Ice & Fire

You will implement an ERC-20 Token.

It will have 2 additional features:

1. An admin can freeze accounts
2. Any user can burn (destroy) their own tokens

Details in Canvas.