Introduction to smart Contracts

Blockchain Business Applications

Aalto University
School of Business

Introducing Smart Contracts

- **Computer programs stored on the blockhain that digitize contracts by converting their terms into computer code that is automatically executed when the terms are met**
  - Follow an “if this then that” logic
  - Behave exactly as programmed and cannot be changed
- **Used to digitally facilitate, verify, and enforce the contracts made between two or more parties**
- **Can also be used to automate workflows by performing next action when conditions are met**
  → Removes the need for human intermediaries

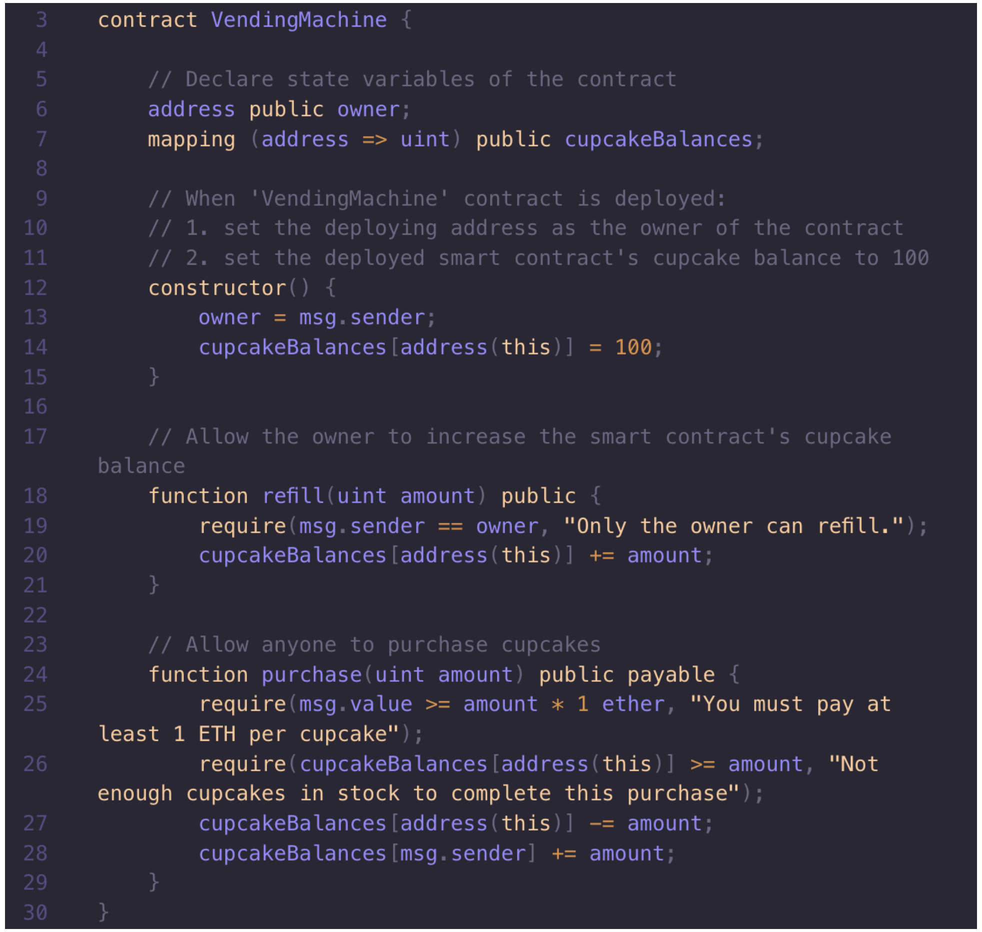
Aalto University School of Business

Sources: Wang et al., 2019, Ethereum: https://ethereum.org/en/smart-contracts/, IBM: https://www.ibm.com/topics/smart-contracts

Vending machine example

- **Often used metaphor of smart contracts is a vending machine**
- **With the right inputs (snack selected + money), a certain output is guaranteed (snack obtained)**
- **Vending machine removes the need for a vendor employee**

→ Similarly, smart contracts can replace intermediaries in many industries



```
contract VendingMachine {

    // Declare state variables of the contract
    address public owner;
    mapping (address => uint) public cupcakeBalances;

    // When 'VendingMachine' contract is deployed:
    // 1. set the deploying address as the owner of the contract
    // 2. set the deployed smart contract's cupcake balance to 100
    constructor() {
        owner = msg.sender;
        cupcakeBalances[address(this)] = 100;
    }

    // Allow the owner to increase the smart contract's cupcake balance
    function refill(uint amount) public {
        require(msg.sender == owner, "Only the owner can refill.");
        cupcakeBalances[address(this)] += amount;
    }

    // Allow anyone to purchase cupcakes
    function purchase(uint amount) public payable {
        require(msg.value >= amount * 1 ether, "You must pay at least 1 ETH per cupcake");
        require(cupcakeBalances[address(this)] >= amount, "Not enough cupcakes in stock to complete this purchase");
        cupcakeBalances[address(this)] -= amount;
        cupcakeBalances[msg.sender] += amount;
    }
}
```

Aalto University
School of Business

Source: https://ethereum.org/en/developers/docs/smart-contracts/

Applications

Blockchain platforms

- Creating and trading digital assets (NFTs)
- Decentralized exchanges for cryptocurrencies

Finance

- Automatic insurance processing and payments (crops, flight delays)
- Dividend payments, stock splits, liability management

Management & Government

- Certifying intellectual properties and digital rights
- Payment systems for work and pension, e-voting and other government services

- **Other application scenarios are e.g., health care and energy industries and emerging technologies such as Internet of Things**

Sources: Wang et al., 2019, Ethereum: https://ethereum.org/en/smart-contracts/, https://ethereum.org/en/get-eth/#dex/ , Etherisc: https://etherisc.com/

Benefits

Speed and efficiency

- Executed immediately once the terms are met, no need to wait for human action
- Digital and automated process reduces paperwork and errors

Trust and transparency

- No third party involved
- Records of transactions shared between parties
- Visible terms and predictable outcomes

Security and privacy

- Encrypted records difficult to hack
- Each record connected to other records on a distributed ledger

Cost savings

- Less human intermediaries reduces also costs related to transactions

Aalto University School of Business

Source: Ethereum: https://ethereum.org/en/smart-contracts/ IBM: https://www.ibm.com/topics/smart-contracts

Challenges

- **Some limitations of blockchain hinder the development of smart contracts**

    1. Irreverrible bugs
    2. Performance issues
    3. Lack of trusted data feeds
    4. Lack of standards and regulations

- **There are also legal issues, such as smart contracts being inconsistent with laws, or posing potential hazards, such as data loss or privacy breach**

Source: Wang et al., 2019,

Further learning opportunites

- This course is about introducing smart contracts and not about codes for smart contracts.
- However. if you are interested in learning about how to develop smart contracts, you can follow the link provided in this slide and learn more about smart contracts on Ethereum:

https://ethereum.org/en/developers/learning-tools/