OST
Eastern Switzerland
University of Applied Sciences

Blockchain (BlCh)

Tokens and NFTs

Thomas Bocek

09.10.2025

Learning Goals

- Lecture 5
  - Knowledge
    - Explain the difference between fungible and non-fungible tokens
  - Skills
    - Implement ERC-20 token contracts
    - Implement ERC-721 NFT contracts with minting functionality
    - Deploy smart contracts to testnets using Remix and Metamask
  - Application
    - Evaluate appropriate token types for specific use cases
    - Design token metadata and URI structures
    - Integrate OpenZeppelin libraries into contract development

Introduction

- Explanation: “Fungible tokens or assets are divisible and non-unique. For instance, fiat currencies like the dollar are fungible: A $1 bill in New York City has the same value as a $1 bill in Miami. A fungible token can also be a cryptocurrency like Bitcoin: 1 BTC is worth 1 BTC, no matter where it is issued.”
- 3 characteristics
  - Interchangeability
    - With other tokens of the **same** type
  - Divisibility
    - Into smaller units
  - Standardization
    - High degree of standardization and uniformity
- Examples
  - Bitcoin, Ethers, USDC, USDT, OST Token
  - Native vs. non-native
    - Native tokens: Built into blockchain protocol (Bitcoin, Ether)
      - Used directly for transaction fees
      - Core part of blockchain infrastructure
    - Non-native tokens: Created via smart contracts (USDC, USDT, OST)
      - Require smart contract interaction
      - Can use ERC-4337 for gas payment

OST

Cryptocurrencies

- Cryptocurrencies are fungible tokens
  - Digital currency not reliant on a central authority
  - CBDC is digital currency backed by central authority
    - “In contrast to cryptocurrencies, a central bank digital currency would be centrally controlled”
- Exchange cryptocurrencies
  - Decentralized exchhanges (DEX) – same chain/cross chain, centralized exchanges (CEX)

- 6 conditions (by Jan Lansky)
  1) The system does not require a central authority, distributed achieve consensus on its state.
  2) The system keeps an overview of cryptocurrency units and their ownership.
  3) The system defines whether new cryptocurrency units can be created. If new cryptocurrency units can be created, the system defines the circumstances of their origin and how to determine the ownership of these new units.
  4) Ownership of cryptocurrency units can be proved exclusively cryptographically.
  5) The system allows transactions to be performed in which ownership of the cryptographic units is changed. A transaction statement can only be issued by an entity proving the current ownership of these units.
  6) If two different instructions for changing the ownership of the same cryptographic units are simultaneously entered, the system performs at most one of them.

OST

Fungible Tokens: Applications and Regulatory Context

- Many types of tokens
  - Stablecoins
    - Pegged to fiat money, e.g., USDC, XCHF
  - Gas for smart contracts
    - Use to execute smart contracts
  - Altcoins
    - Non mainstream coins (aka sh*tcoins), definition is vague
    - Speculation...
- Swiss DLT law, since 01.08.2021
  - Licence for DLT trading facilities
  - Electronic registers
- Finma categorization (payment, utility and asset tokens):
  - Payment Tokens: Pure cryptocurrencies (no securities regulation)
  - Utility Tokens: Access to services (lighter regulation if genuine utility)
  - Asset Tokens: Represent real assets (full securities regulation applies)
    - Prospectus requirement
    - Exception: Physical assets without participation rights
  - No-Action Letter: couple of months, provides regulatory clarity

OST

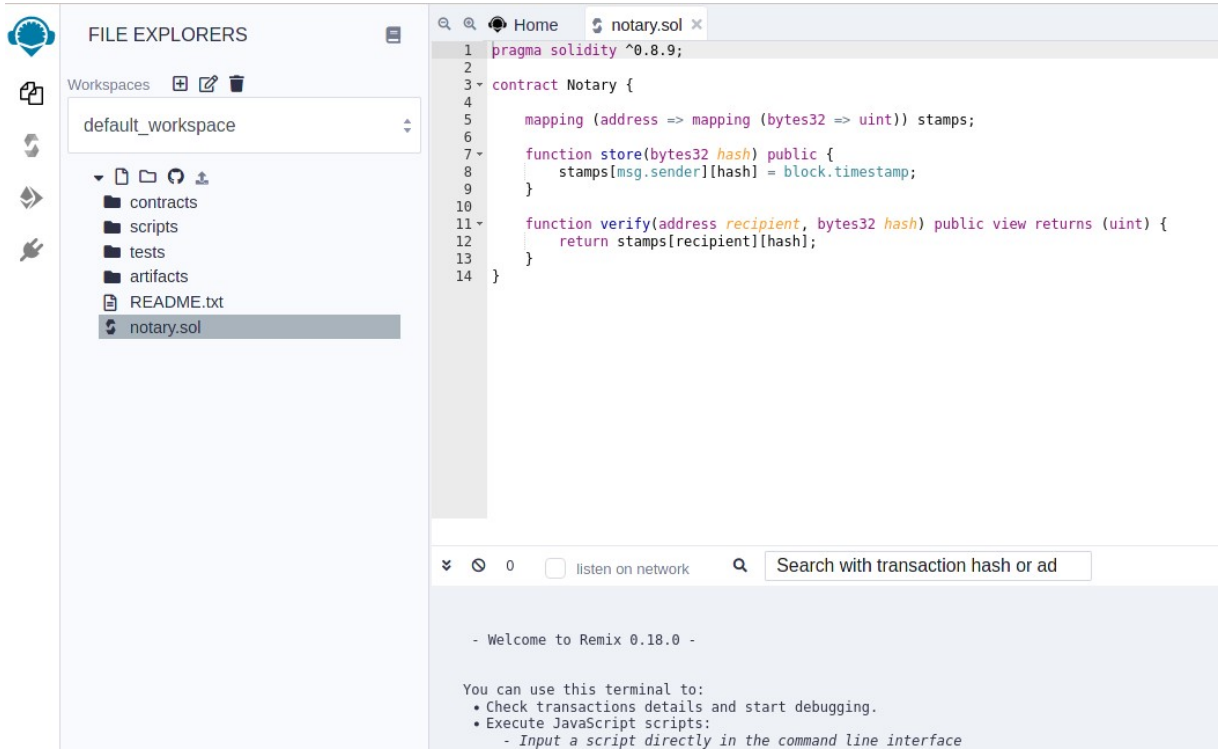
FT / Tokens / ERC-20

- ERC-20
  - Token contract keeps track of fungible tokens
  - Can be used as vault for NFTs
- Optional (highly recommended)
  - `function name() public view returns (string)`
  - `function symbol() public view returns (string)`
  - `function decimals() public view returns (uint8)`
- Events
  - `event Transfer(address indexed _from, address indexed _to, uint256 _value)`
  - `event Approval(address indexed _owner, address indexed _spender, uint256 _value)`
- Mandatory
  - `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)`
    - Direct token transfer from your address
  - `function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)`
    - Allows third party (e.g., DEX) to transfer tokens on your behalf after approval
  - `function approve(address _spender, uint256 _value) public returns (bool success)`
    - Grants permission for transferFrom operations
  - `function allowance(address _owner, address _spender) public view returns (uint256 remaining)`

OST

ERC-20 Implementation

- OpenZeppelin – many default contracts, very good source
  - Let’s implement a simple ERC20 and deploy on testnet
    - Remix IDE



OST