OST
Eastern Switzerland
University of Applied Sciences

Blockchain (BlCh)

Tokens and NFTs

Thomas Bocek

09.10.2025

NFT

- Non-fungible token

fungible *adj.* [LAW] — ersetzbar

- Unique token on a public blockchain
  - Guarantees that a digital asset is unique and not interchangeable
  - Can be any digital data that can be hashed (only hash is stored on-chain)
  - With NFT: proof of ownership (you can copy the digital data, but the ownership remains)
- NFT market place OpenSea - open to everyone, for any NFT

CryptoPunk #7523, sold for 11.8m USD [link]

OST

NFT

- Also big players: NBA
  - Fan Tokens are not NFTs, e.g., PSG, FCB, YB
    - Six-month high as 2024
- Market Evolution
  - 2021: Biggest NFT sale so far: Beeple sold for ~$70m, Everydays: the First 5000 Days
  - Current: Post-boom consolidation
  - Shift from speculation to utility (tickets, gaming, authentication)
  - B.20 token case study
    - NFT (also Beeples) stored in B.20 vault, where artists are also participating – conflict of interest?
    - B.20 went up to 25$, now down to 0.05$
- Collectible media (football, basketball players), eg., video, Jack Dorsey sold first twitter post for $2.9m
- Tickets, every entry ticket is unique, keep ticket after events, fee for artist on reselling NFT ticket
- Artists sell music as NFT
- Movies: limited-edition Deadpool 2 digital posters
- Memes: e.g., disaster girl, nyan cat
- Fashion: add NFT token to make clothes unique (sneakers)
- Source code: Tim Berners Lee, $5.4m
- Many use cases not relevant today
  - NFT items in games, e.g., CS:GO skins (traded decentrally, but not an NFT)

OST

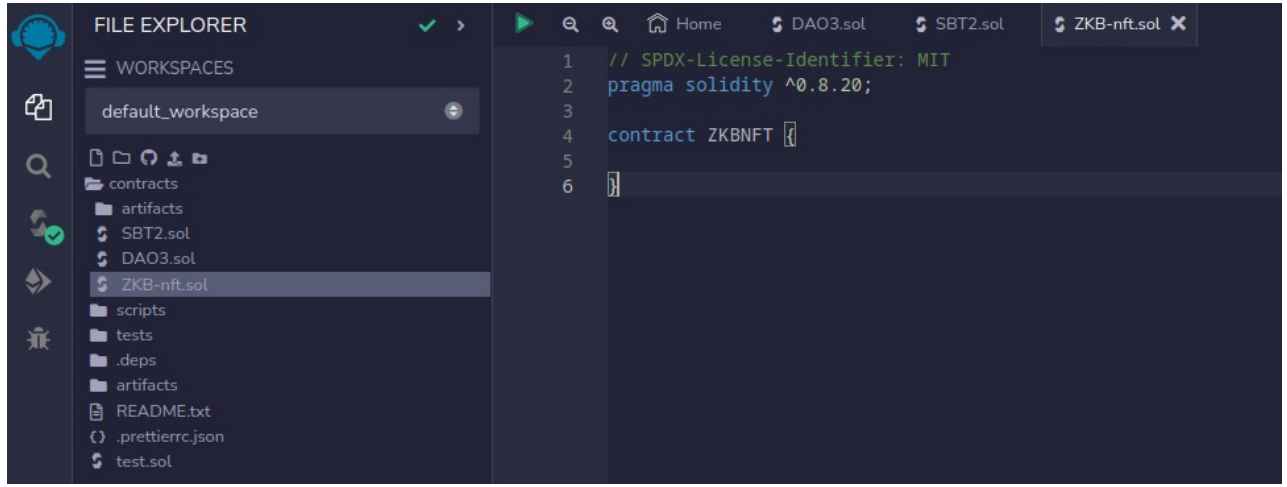
ERC-721

- Non-Fungible Token Standard
  - June 2018 - ERC721 accepted as final
  - “There is either one NFT or there are none”
- Pros
  - Enables peer-to-peer trading of unique digital assets without intermediaries, reducing transaction costs and increasing accessibility.
  - Operates 24/7 on decentralized marketplaces (e.g., OpenSea), ensuring continuous trading and transparency.
  - Provides immutable proof of ownership for digital assets through blockchain records, eliminating disputes over authenticity.
- Cons
  - Requires significant technical expertise to implement securely, increasing development complexity for creators.
  - Ownership of an NFT does not automatically transfer copyright or usage rights
    - NFT ownership ≠ Copyright ownership
    - Buying NFT = proof of ownership of token
    - Copyright remains with creator unless explicitly transferred
    - Most NFT sales do NOT include commercial usage rights
  - Limited to digital assets by default, though bridges to physical-world tokenization (e.g., event tickets, real estate) are emerging.

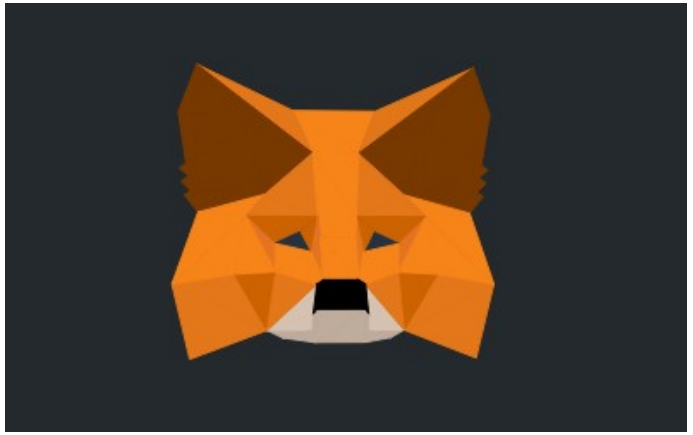
OST

NFT Implementation – Mandatory Interfaces und Events

- On Ethereum, OpenZeppelin has many standard implemented, for IDE Remix is a good choice
- Let’s implement and deploy an NFT



- Metamask
  - Crypto wallet and gateway to blockchain apps, available as a browser extension



- We’ll use that to connect the browser to the Ethereum network

OST

Simple ERC-721

- Almost empty contract - set meta data
  - Required / optional functions

```
contract BlChNFT is IERC721 {
    function name() external pure returns (string memory){return "Blockchain NFT";}
    function symbol() external pure returns (string memory){return "BlchNFT";}
    function tokenURI(uint256 _tokenId) external view returns (string memory){return
                                        string.concat("https://dsl.i.ost.ch/", Strings.toString(_tokenId))}
    function balanceOf(address _owner) external view returns (uint256) {return 1;}
    function ownerOf(uint256 _tokenId) external view returns (address) {return address(0);}
    function safeTransferFrom(address _from, address _to, uint256 _tokenId, bytes memory data) external override {}
    function safeTransferFrom(address _from, address _to, uint256 _tokenId) external override {}
    function transferFrom(address _from, address _to, uint256 _tokenId) external override{}
    function approve(address _approved, uint256 _tokenId) external override{}
    function setApprovalForAll(address _operator, bool _approved) external{}
    function getApproved(uint256 _tokenId) external view returns (address){return address(0);}
    function isApprovedForAll(address _owner, address _operator) external view returns (bool){return true;}
    function supportsInterface(bytes4 interfaceID) external view returns (bool){return true};
}
```


OST

Simple ERC-721

- Lets use the Imports this time
- Strings.sol
- Address.sol

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.9;

import "@openzeppelin/contracts/token/ERC721/IERC721.sol";
import "@openzeppelin/contracts/token/ERC721/extensions/IERC721Metadata.sol";


contract BlChNFT is IERC721 {

}
```

```
import "@openzeppelin/contracts/utils/Strings.sol";
…
string.concat("https://dsl.i.ost.ch/", Strings.toString(_tokenId))
…
```


OST

Simple ERC-721

- **No checks!**
- State variables
- Implement view

```
// Mapping owner address to token count
mapping(address => uint256) private _balances;
// Mapping from token ID to owner address
mapping(uint256 => address) private _owners;
// Mapping from token ID to approved address
mapping(uint256 => address) private _tokenApprovals;
// Mapping from owner to operator approvals
mapping(address => mapping(address => bool)) private _operatorApprovals;
```

```
function balanceOf(address _owner) external view returns (uint256) {
    return _balances[_owner];
}
function ownerOf(uint256 _tokenId) external view returns (address) {
    return _owners[_tokenId];
}
function getApproved(uint256 _tokenId) external view returns (address){
    return _tokenApprovals[_tokenId];
}
function isApprovedForAll(address _owner, address _operator) external view returns (bool){
    return _operatorApprovals[_owner][_operator];
}
```


OST

Simple ERC-721

- TransferFrom
- SafeTransferFrom

```
function transferFrom(address _from, address _to, uint256 _tokenId) public
override{
    address owner = ownerOf(_tokenId);
    require(msg.sender == owner || getApproved(_tokenId) == msg.sender ||
isApprovedForAll(owner, msg.sender), "ERC721: safeTransferFrom caller is not
owner nor approved for all");
    // Clear approvals from the previous owner
    _tokenApprovals[_tokenId] = address(0);
    emit Approval(owner, address(0), _tokenId);

    _balances[_from] -= 1;
    _balances[_to] += 1;
    _owners[_tokenId] = _to;

    emit Transfer(_from, _to, _tokenId);
}
```

```
function safeTransferFrom(address _from, address _to, uint256 _tokenId, bytes
memory _data) public override {
    transferFrom(_from, _to, _tokenId);
    if (_to.isContract()) {
        bytes4 retVal = IERC721Receiver(_to).onERC721Received(msg.sender,
_from, _tokenId, _data);
        require (retVal == IERC721Receiver.onERC721Received.selector, "ERC721:
transfer to non ERC721Receiver implementer");
    }
}
```


OST

Simple ERC-721

- Let’s mint!

```
address private ownerNFT;
constructor() {
    ownerNFT = msg.sender;
}
function _mint(address to, uint256 tokenId) external {
    require(ownerNFT == msg.sender, "only onwner");
    require(_owners[tokenId] == address(0), "ERC721: token already minted");

    _balances[to] += 1;
    _owners[tokenId] = to;
    emit Transfer(address(0), to, tokenId);
}
```


OST

Extensions

- Optional Enumaration - discoverable
  - `function totalSupply() external view returns (uint256);`
  - `function tokenByIndex(uint256 _index) external view returns (uint256);`
  - `function tokenOfOwnerByIndex(address _owner, uint256 _index) external view returns (uint256);`
- Mandatory ERC-165
  - `function supportsInterface(bytes4 interfaceID) external view returns (bool);`
    - `balanceOf(address) → 0x70a08231`
- Optional ERC721TokenReceiver
  - Returns 0x150b7a02 (IERC721Receiver.onERC721Received.selector)
- Optional ERC721Metadata
- OpenSea
  - More metadata here
- ERC 1155
  - Supports fungible and non-fungible tokens in a single contract, reducing deployment costs and simplifying asset management
  - Enables batch transfers of multiple tokens in one transaction, ideal for mass distributions (e.g., game items)
  - But: complexity

OST