OST
Eastern Switzerland
University of Applied Sciences

Implementation and Deployment of Smart Contracts

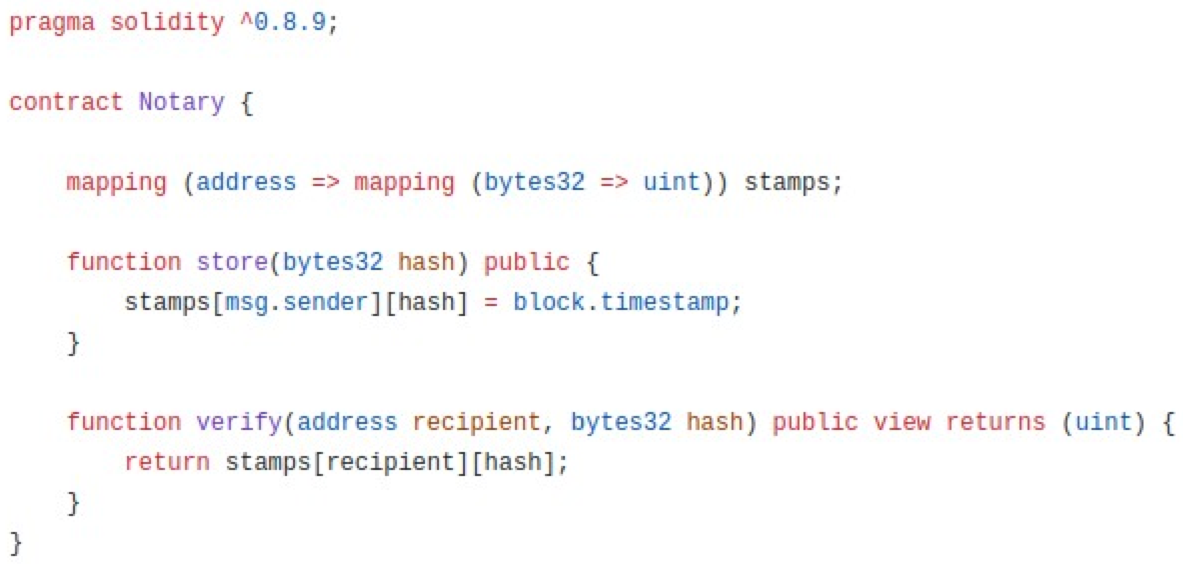
Thomas Bocek

21.10.2024

How to Implement a Smart Contract

- What is a Smart Contract
  - Program executed within a blockchain context
  - Read blockchain data, execution part of consensus – everyone comes to the same result
- Dedicated blockchain programming languages: Solidity, Michelson, Move, Chaincode, Vyper, Plutus
  - Existing languages: NEO uses C#, Java, Solana uses Rust, more generic: WebAssembly: Polkadot, DFINITY

- Example Solidity for Ethereum

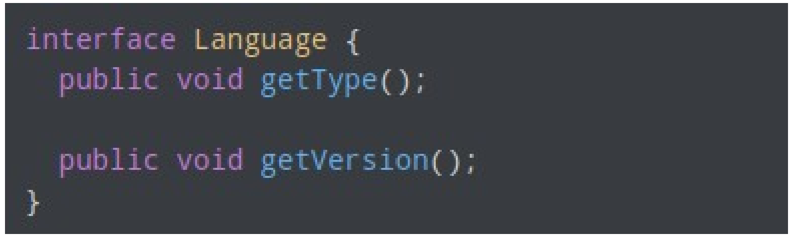


```
pragma solidity ^0.8.9;

contract Notary {

    mapping (address => mapping (bytes32 => uint)) stamps;

    function store(bytes32 hash) public {
        stamps[msg.sender][hash] = block.timestamp;
    }

    function verify(address recipient, bytes32 hash) public view returns (uint) {
        return stamps[recipient][hash];
    }
}
```

https://github.com/tbocek/FS21/blob/main/ethereum/Notary.sol

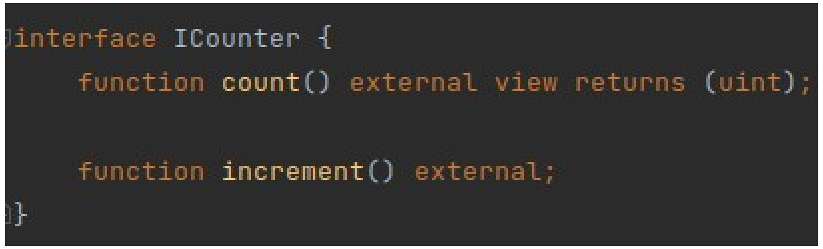
How to Implement a Smart Contract

- How can an app know how to interact with a smart contract? (e.g., this DeFi app?)
  - Build everything yourself
  - Use standards / interfaces
- Interface in Java



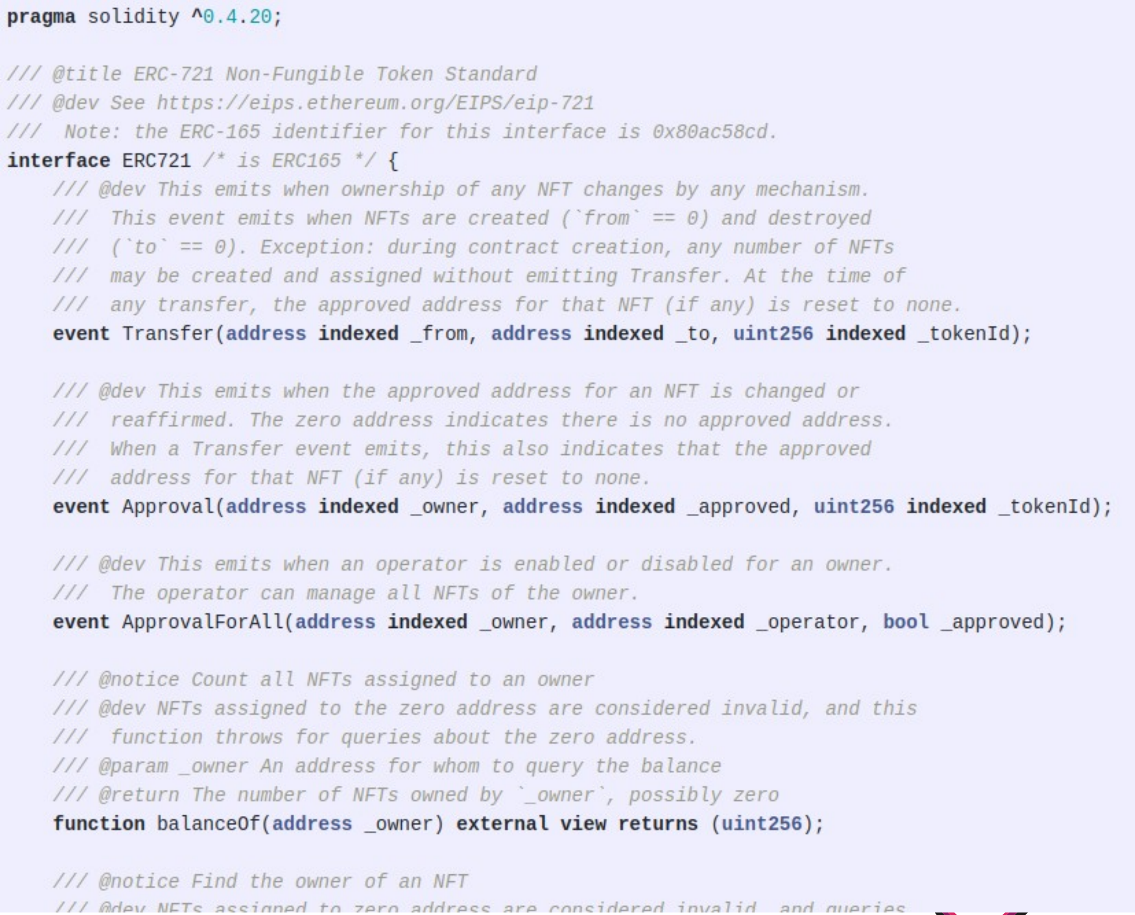
```
interface Language {
  public void getType();

  public void getVersion();
}
```

- Interface in Solidity



```
interface ICounter {
    function count() external view returns (uint);

    function increment() external;
}
```

- ERC (Ethereum Request for Comments)
  - Defines interfaces, e.g.,



```
pragma solidity ^0.4.20;

/// @title ERC-721 Non-Fungible Token Standard
/// @dev See https://eips.ethereum.org/EIPS/eip-721
///  Note: the ERC-165 identifier for this interface is 0x80ac58cd.
interface ERC721 /* is ERC165 */ {
    /// @dev This emits when ownership of any NFT changes by any mechanism.
    ///  This event emits when NFTs are created (`from` == 0) and destroyed
    ///  (`to` == 0). Exception: during contract creation, any number of NFTs
    ///  may be created and assigned without emitting Transfer. At the time of
    ///  any transfer, the approved address for that NFT (if any) is reset to none.
    event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);

    /// @dev This emits when the approved address for an NFT is changed or
    ///  reaffirmed. The zero address indicates there is no approved address.
    ///  When a Transfer event emits, this also indicates that the approved
    ///  address for that NFT (if any) is reset to none.
    event Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId);

    /// @dev This emits when an operator is enabled or disabled for an owner.
    ///  The operator can manage all NFTs of the owner.
    event ApprovalForAll(address indexed _owner, address indexed _operator, bool _approved);

    /// @notice Count all NFTs assigned to an owner
    /// @dev NFTs assigned to the zero address are considered invalid, and this
    ///  function throws for queries about the zero address.
    /// @param _owner An address for whom to query the balance
    /// @return The number of NFTs owned by `_owner`, possibly zero
    function balanceOf(address _owner) external view returns (uint256);

    /// @notice Find the owner of an NFT
    /// @dev NFTs assigned to zero address are considered invalid, and queries
```

NFT

- Non-fungible token

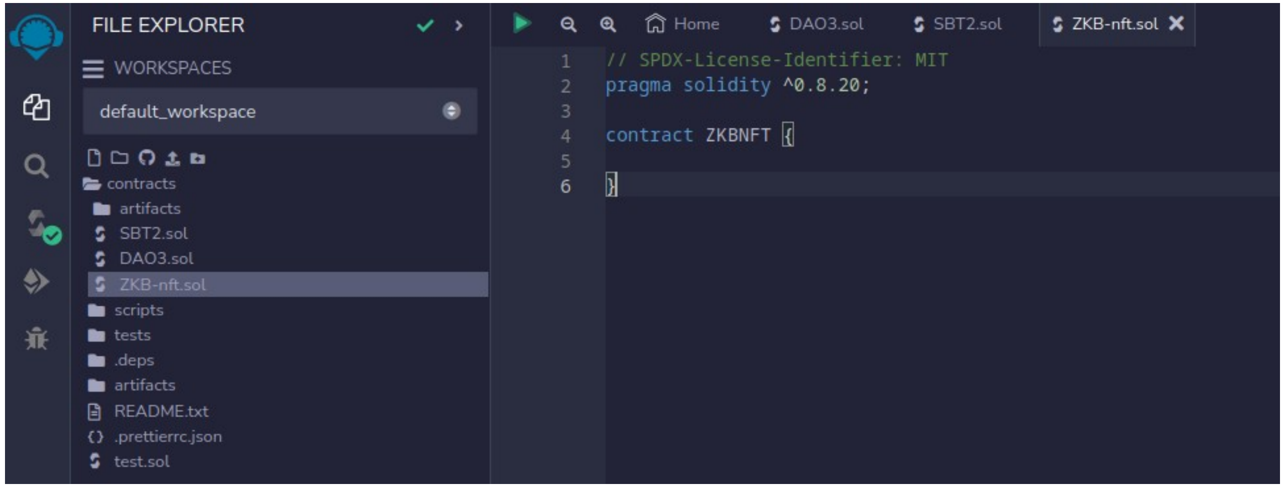
fungible *adj.* [LAW] ersetzbar

- 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]

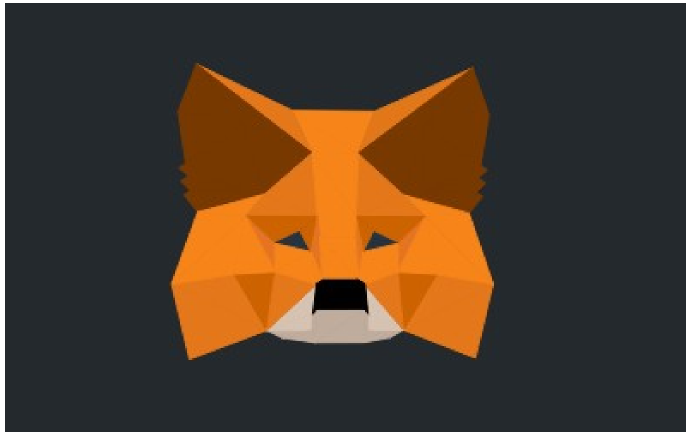
NFT Implementation – Mandatory Interfaces und Events

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



```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.20;

contract ZKBNFT {

}
```

- 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