NPTEL ONLINE CERTIFICATION COURSES

Blockchain and its applications

Bishakh Chandra Ghosh

Department of Computer Science & Engineering
Indian Institute of Technology Kharagpur

Lecture 25: Ethereum 4

CONCEPTS COVERED

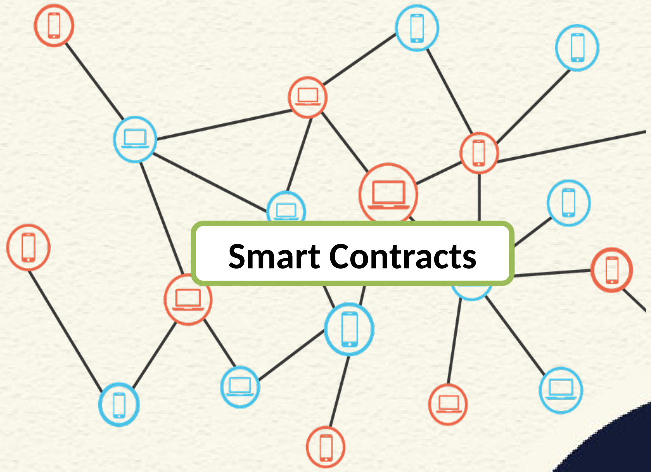
- Ethereum smart contracts
- Ethereum Virtual Machine (EVM)
- Solidity language
- Deploy and execute contracts

KEYWORDS

- Ethereum Smart Contracts
- EVM
- Solidity

Smart Contracts

- Program that is executed in a decentralized setting.
- Acts on distributed ledger data.
- Output of the program depends on consensus of independent participants executing it.



Ethereum Smart Contracts

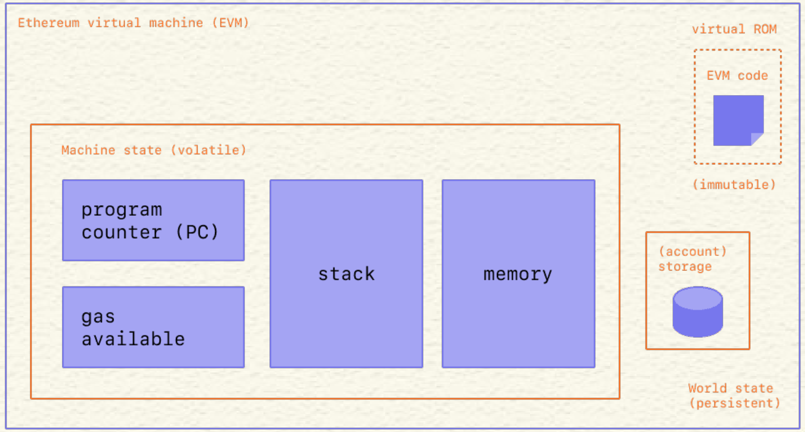
- Program that reads data from the Ethereum ledger, performs operations on it, and writes back the output to the ledger.

- Smart contracts are a type of Ethereum account.
  - **have a balance**
  - **can send transactions over the network**
  - not controlled by a user, run as programmed

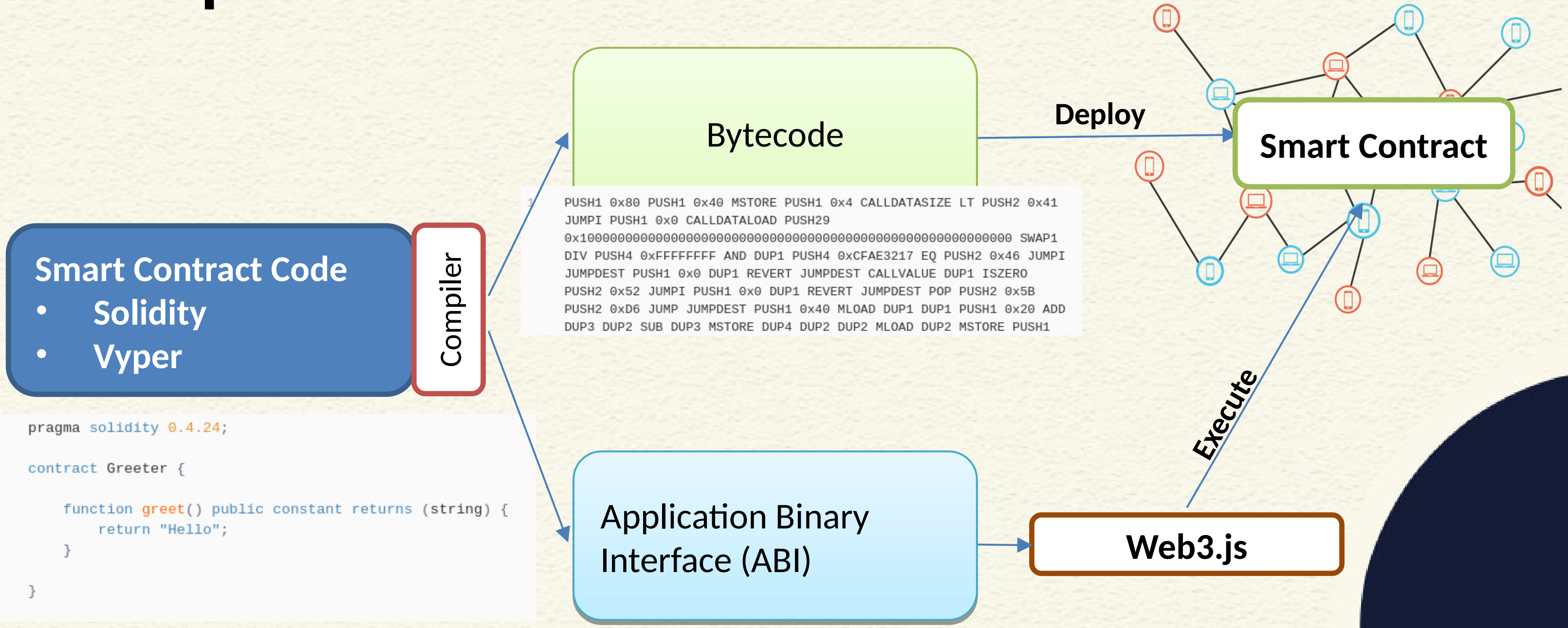
- User accounts interact with a smart contract by **submitting transactions** that **execute a function** defined on the smart contract.

Ethereum Virtual Machine - EVM

- Ethereum implements a **distributed state machine / replicated state machine.**
- Ledger data holds the state - accounts, balances, and other variables.
- Transactions deterministically change the machine from one state to another.



Compiler



```
PUSH1 0x80 PUSH1 0x40 MSTORE PUSH1 0x4 CALLDATASIZE LT PUSH2 0x41
JUMPI PUSH1 0x0 CALLDATALOAD PUSH29
0x100000000000000000000000000000000000000000000000000000000 SWAP1
DIV PUSH4 0xFFFFFFFF AND DUP1 PUSH4 0xCFAE3217 EQ PUSH2 0x46 JUMPI
JUMPDEST PUSH1 0x0 DUP1 REVERT JUMPDEST CALLVALUE DUP1 ISZERO
PUSH2 0x52 JUMPI PUSH1 0x0 DUP1 REVERT JUMPDEST POP PUSH2 0x5B
PUSH2 0xD6 JUMP JUMPDEST PUSH1 0x40 MLOAD DUP1 DUP1 PUSH1 0x20 ADD
DUP3 DUP2 SUB DUP3 MSTORE DUP4 DUP2 DUP2 MLOAD DUP2 MSTORE PUSH1
```

```
pragma solidity 0.4.24;

contract Greeter {

    function greet() public constant returns (string) {
        return "Hello";
    }

}

```

https://vyper.readthedocs.io/en/stable/
https://docs.soliditylang.org/en/v0.8.9/

Solidity

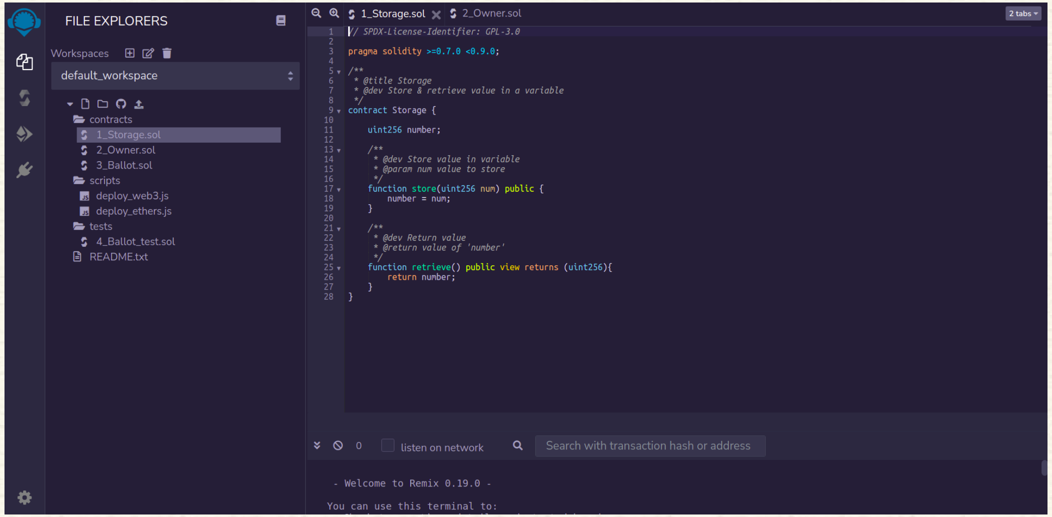
- High-level language for implementing smart contracts
  - Statically typed
  - Supports inheritance
  - Libraries
  - Complex user-defined types
- Syntax influenced by Javascript and C++

```
// SPDX-License-Identifier: GPL-3.0
pragma solidity >=0.4.16 <0.9.0;

contract SimpleStorage {

uint storedData;

function set(uint x) public {
storedData = x;
 }

function get() public view returns (uint) {
 return storedData;
}

}
```

https://docs.soliditylang.org/en/v0.8.9/

Compiling Solidity

- **solc compiler**
  - Compile .sol files to bytecode
  - Command line tool
- **Remix editor**
  - Browser based
  - Compile, emulate, deploy contracts
  - https://remix.ethereum.org/



```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

/**
 * @title Storage
 * @dev Store & retrieve value in a variable
 */
contract Storage {

    uint256 number;

    /**
     * @dev Store value in variable
     * @param num value to store
     */
    function store(uint256 num) public {
        number = num;
    }

    /**
     * @dev Return value
     * @return value of 'number'
     */
    function retrieve() public view returns (uint256){
        return number;
    }
}
```

```
 - Welcome to Remix 0.19.0 -

You can use this terminal to:
```

https://docs.soliditylang.org/en/v0.8.9/installing-solidity.html#installing-solidity

Simple Storage Contract

```
// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <0.9.0;

contract Storage {

        uint256 positivenumber;

        function store(uint256 inputnumber) public {
                positivenumber = inputnumber;
        }

        function readdata() public view returns (uint256){
                return positivenumber;
        }

}
```

Executing with Web3

```
var Web3 = require('web3');
var Contract = require('web3-eth-contract');

Contract.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'));

var myContract = new Contract(<ABI>,

"0xb3f36458FFc0C686DB4f2FF6002a55bFD85C03C8",
{
from: "0xd84a0607843b28c3f468857f82f784d9ff743bf8",
gasPrice: "20000000000"
}

);


myContract.methods.readdata().call().then(function (output)
{ console.log(output) });
```

Read data

Executing with Web3

```
var Web3 = require('web3');
var Contract = require('web3-eth-contract');

Contract.setProvider(new Web3.providers.HttpProvider('http://localhost:8545'));

var myContract = new Contract(<ABI>,

"0xb3f36458FFc0C686DB4f2FF6002a55bFD85C03C8",
{
from: "0xd84a0607843b28c3f468857f82f784d9ff743bf8",
gasPrice: "20000000000"
}

);


myContract.methods.store("11").send().then(function (output)
{ console.log(output) });
```

Write data

Conclusion

- Ethereum smart contracts
- Remix editor
- Executing smart contracts from Web3 for DAPPS

Thank you