TinTin

以太坊入门第二课 Office Hour

助教：吕宁

1、课程回顾

- 一笔交易的背后组成：rawTransaction
- ABI编码：合约间调用或发送消息时的消息格式。来定义函数签名，参数编码，返回结果编码等
- EVM：合约执行引擎
- Solidity基础入门

rawTransaction

- https://goerli.etherscan.io/tx/0x9bba098fcc7bfac3419c958cd70b834519710e5f3cb994bf0927959be60872dc
- curl https://eth-goerli.g.alchemy.com/v2/n34clhl26p2RYu91dpwueZTAm7YSfMu3 -X POST -H "Content-Type: application/json" -d '{"jsonrpc":"2.0","method":"eth_getTransactionByHash","params":["0x9bba098fcc7bfac3419c958cd70b834519710e5f3cb994bf0927959be60872dc"],"id":1}'
- https://web3js.readthedocs.io/en/v1.9.0/web3-eth.html#gettransaction
- https://docs.ethers.org/v5/api/providers/provider/#Provider-getTransaction

ABI编码：web2.0和web3.0的桥梁

- https://goerli.etherscan.io/tx/0x66a1cb3bf3d0614f207c304446b259890b1828789a95382e45834141a5efaf49
- https://emn178.github.io/online-tools/keccak_256.html 、web3.utils.sha3(“withdraw(uint256)”)
- https://calldata-decoder.apoorv.xyz/
- https://remix.ethereum.org/

EVM：分布式状态机

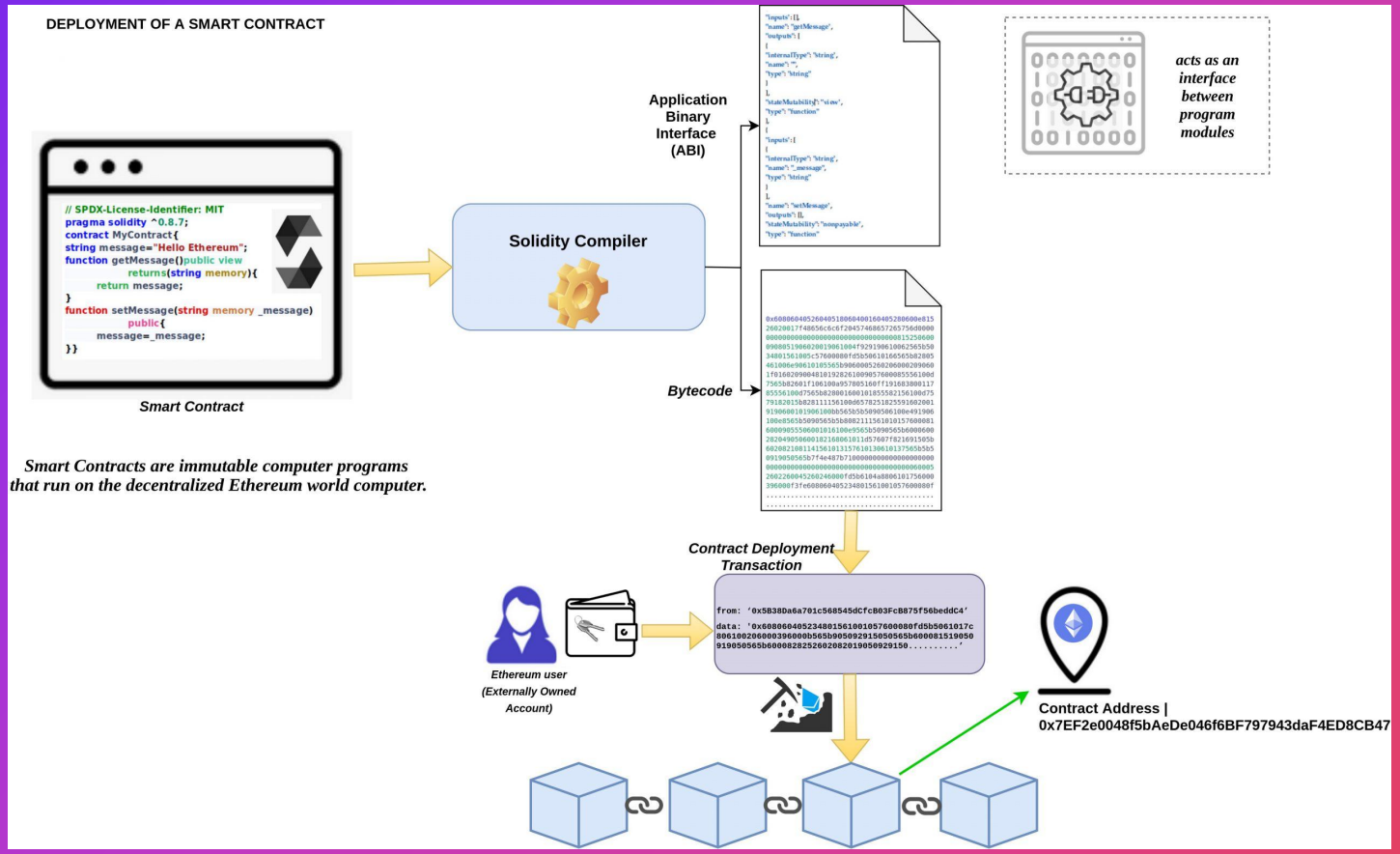
- Y(S, T)= S'
- 交易触发、状态是一个巨大的数据结构Merkle Patricia Tree、给定输入产生确定输出
- https://www.evm.codes/

入门Solidity

- 状态变量、函数、修饰器、事件、结构类型、枚举类型

TINTINLAND

TinTin



Smart Contracts are immutable computer programs that run on the decentralized Ethereum world computer.

2、作业点评

交易哈希:

```
0xefd91225fa06b4de166c6f9b31cf452bf888c40193a9b8db0ba96e0832df4da9
```

原始信息:

```
0x02f87205808459682f008503b80dc50f825208940e55ad64ce78a71687fd3f32dccca941ebeb3d7a87b1a2bc2ec5
//在ethsacn中可以直接获得
```

还原代码:

```
from web3 import Web3
import json
web3 = Web3(Web3.HTTPProvider('https://goerli.infura.io/v3/######################'))
tx_hash ="0xefd91225fa06b4de166c6f9b31cf452bf888c40193a9b8db0ba96e0832df4da9"
tx = web3.eth.get_transaction(tx_hash)
print(tx)
//get_transaction(byte32 txHash) return 交易信息后自动解码
```

还原结果:

```
AttributeDict(
{'accessList': [],
'blockHash':HexBytes('0xbbe908a4bd719d85dd77f3c5c2a4a7670b721edea632bc55e31e8528ec9084fb'),
'blockNumber': 8634833,
'chainId': 5,
'from': '0xfA4E41A12B5f92A223eDBB62601899B39cF6334c',
'gas': 21000,
'gasPrice': 12938010956, 'hash': HexBytes('0xefd91225fa06b4de166c6f9b31cf452bf888c40193a9b8db0ba96e
'input': '0x',
'maxFeePerGas': 15972812047,
'maxPriorityFeePerGas': 1500000000, 'nonce': 0,
'r': HexBytes('0xc239d394b206a1db82fce4656eb9d53d00f9172eb831b5590f253695c9800b45'),
's': HexBytes('0x2a7038584acba9b866bda56c11a33b01835a5171b2473c2649443e07f5a0870d'),
'to': '0x0E55Ad64CE78a71687FD3F32DccCA941eBEB3D7A',
'transactionIndex': 24,
'type': 2,
'v': 1,
'value': 500000000000000000})
```

```
},
{"internalType":"uint256","name":"deadLine","type":"uint256"}
],"name":"mixSwap","outputs":
[{"internalType":"uint256","name":"returnAmount","type":"uint
256"}],"stateMutability":"payable","type":"function"},
{"stateMutability":"payable","type":"receive"}]
```

Decode

Output:

```
{
  "name": "dodoMutliSwap",
  "params": [
    {
      "name": "fromTokenAmount",
      "value": "10000000000000000",
      "type": "uint256"
    },
    {
      "name": "minReturnAmount",
```

3、Q&A

C语言中，有一个main函数，规定了代码从上至下的执行顺序，那合约被调用后，是怎样一个执行的逻辑？