交易与Gas

七哥
https://x.com/0xqige

交易

以太坊是一台状态机，通过一条条消息的有序执行来不断改变状态的

交易本质是传递一条消息：Message（sender,to,input,value,gas）

Output = Call(sender,to,input, value,gas)


```
{
  "action": {
    "from": "0xc733434bf8db449ca95087f1ba5d4b887e3b6651",
    "callType": "staticcall",
    "gas": "0x258ba",
    "input": "0x70a08231000000000000000000000000c733434bf8db449ca95087f1ba5d4b887e3b6651",
    "to": "0x4d0528598f916fd1d8dc80e5f54a8feedcfd4b18",
    "value": "0x0"
  },
  "blockHash": "0x67b80be36b9944e06c76ad4db40bf98958c47970efe1de0e09cc067ed569873b",
  "blockNumber": 20359683,
  "result": {
    "gasUsed": "0x265",
    "output": "0x000000000000000000000000000000000000000000000019ca8365847640b7bffb"
  },
  "subtraces": 0,
  "traceAddress": [
    6,
    3
  ],
  "transactionHash": "0x354b30d1ef16ea6e1a2e8218aeb34e19a72e08e87b84dd03f821cc6ebe4c1e18",
  "transactionPosition": 2,
  "type": "call"
},
```

交易

交易本质是传递一条消息：Message（sender,to,input,value,gas）

- 张三转账 10 ETH 给李四

  MSG（张三，李四，“”, 10ETH，gas）

- 张三转账 1000 USDT 给李四

  MSG（张三，USDT，abi.encode(“transfer(address,uint256)”,李四, 1000), 0ETH，gas）

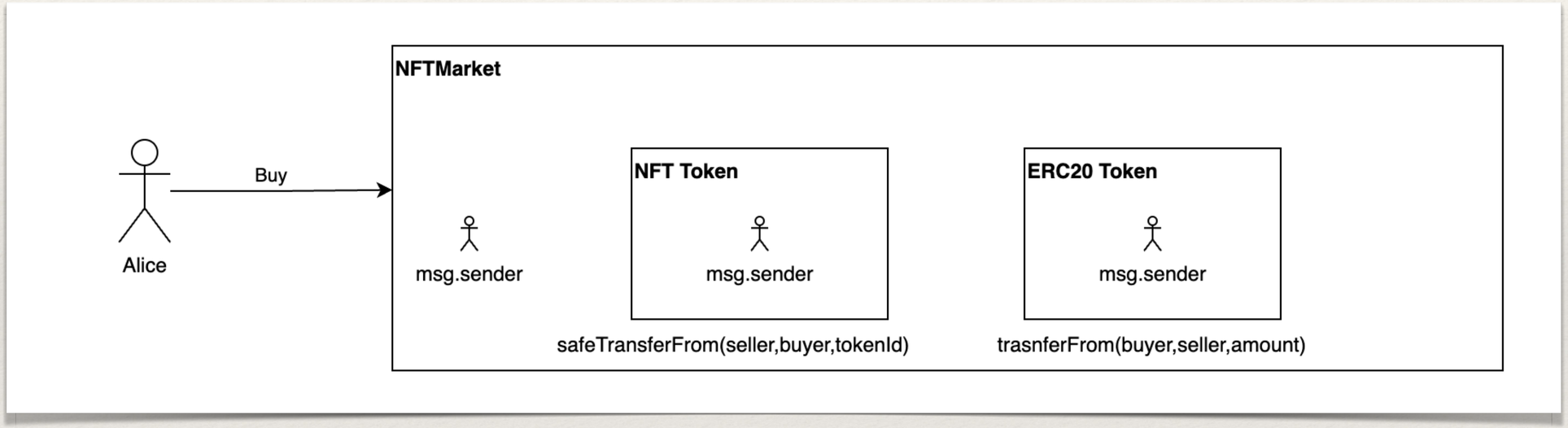
- 王五部署一个新合约

  MSG（王五，0x0，code, 0ETH，gas）

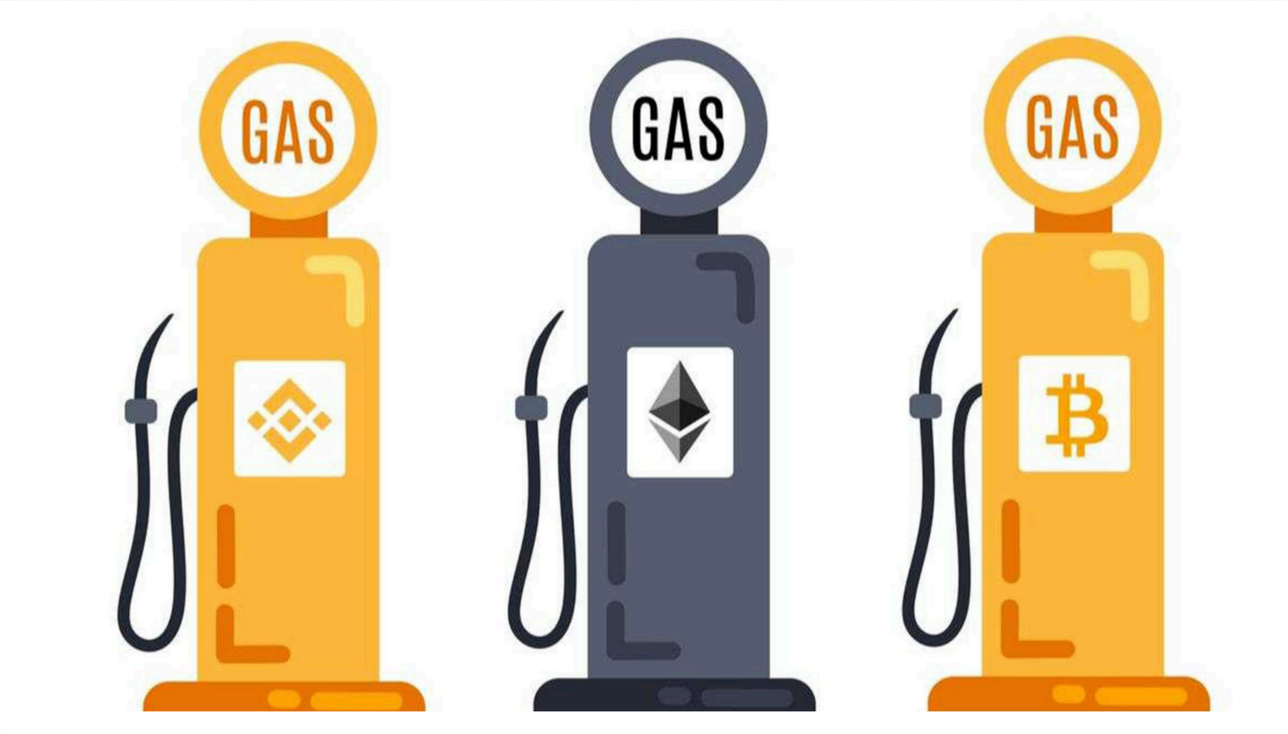
- 王五在NFTMarket 合约中买入NFT

  MSG（王五，NFTMarket，abi.encode(“buy(…)”,…), 0ETH，gas）

msg.sender 是谁？



Gas



Gas： 支付给矿工将交易TX写入区块的好处费，按写入计算量付费

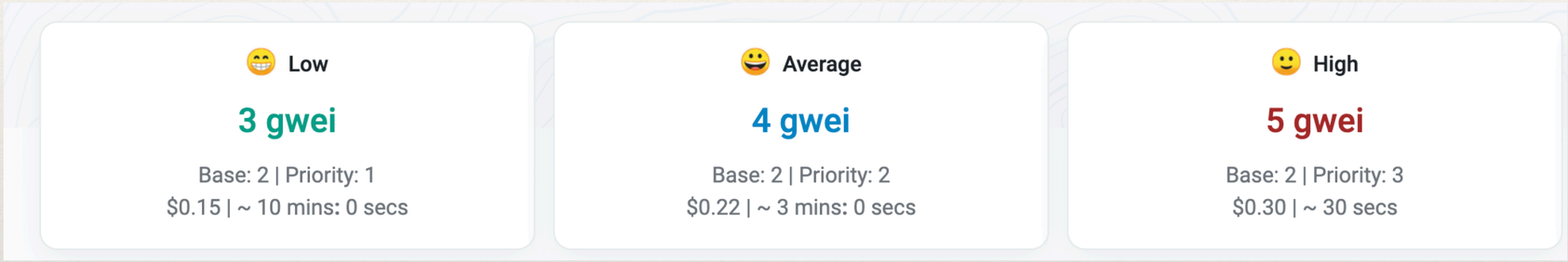
Gas

GasLimit	GasUsed	GasPrice
交易计算量的安全上限	交易实际计算量	愿意为每单位计算量支付的好处费

最终为交易支付的好处费 = GasUsed * GasPrice
并且不会超过 GastLimit * GasPrice

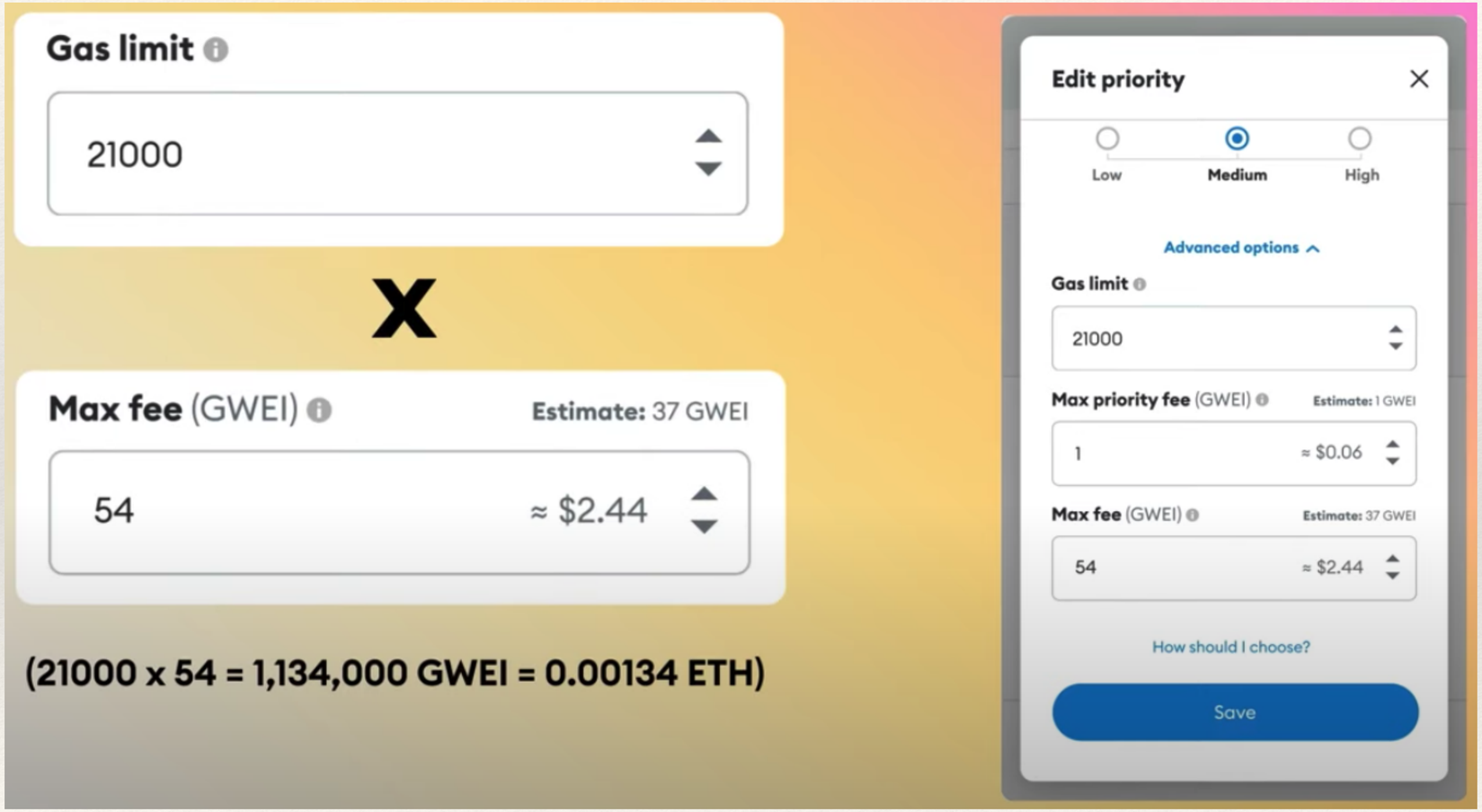
Gas

要处理的交易非常多，因此矿工为了利益最大化，谁支付的 GasPrice 越高，就越先处理谁的交易



拍卖式

Gas - EIP1559

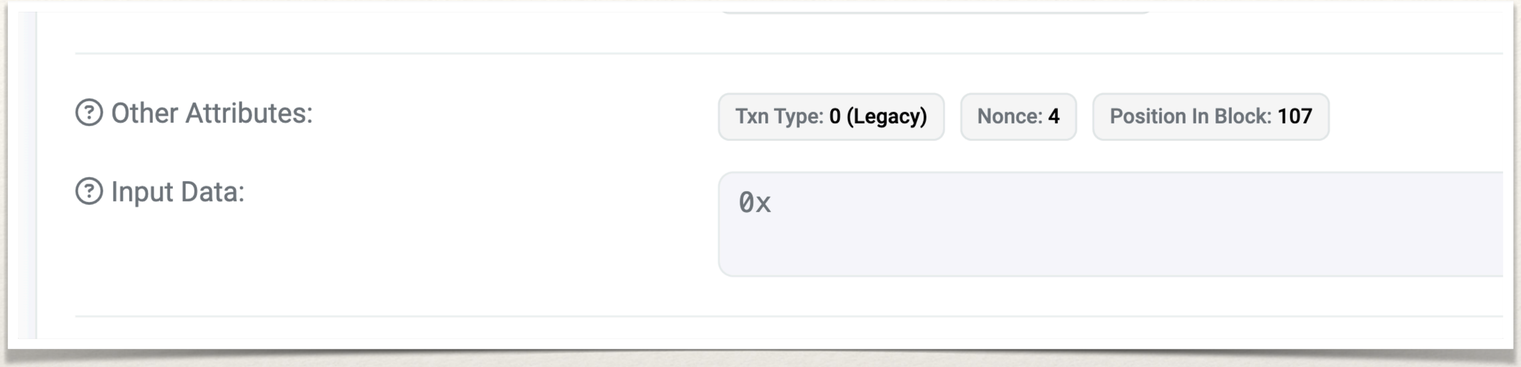


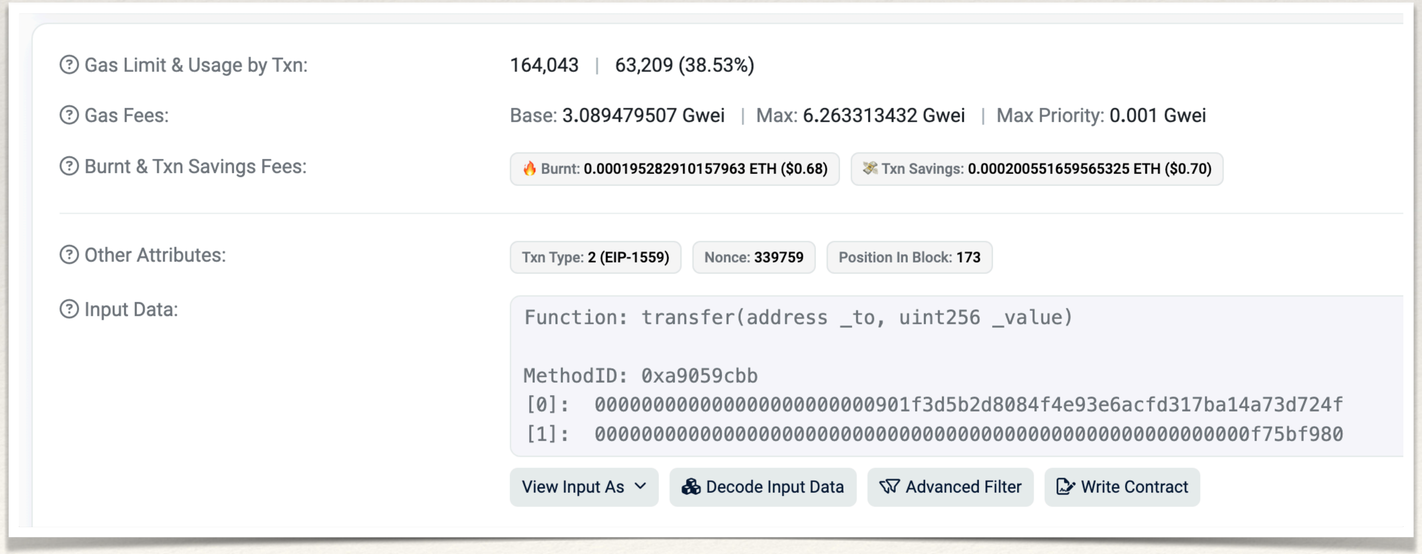
Max priority fee: 愿意给矿工的小费上限

Max fee: 愿意支出的费用，即gasPrice

Price =Base Fee ＋ Priority Fee

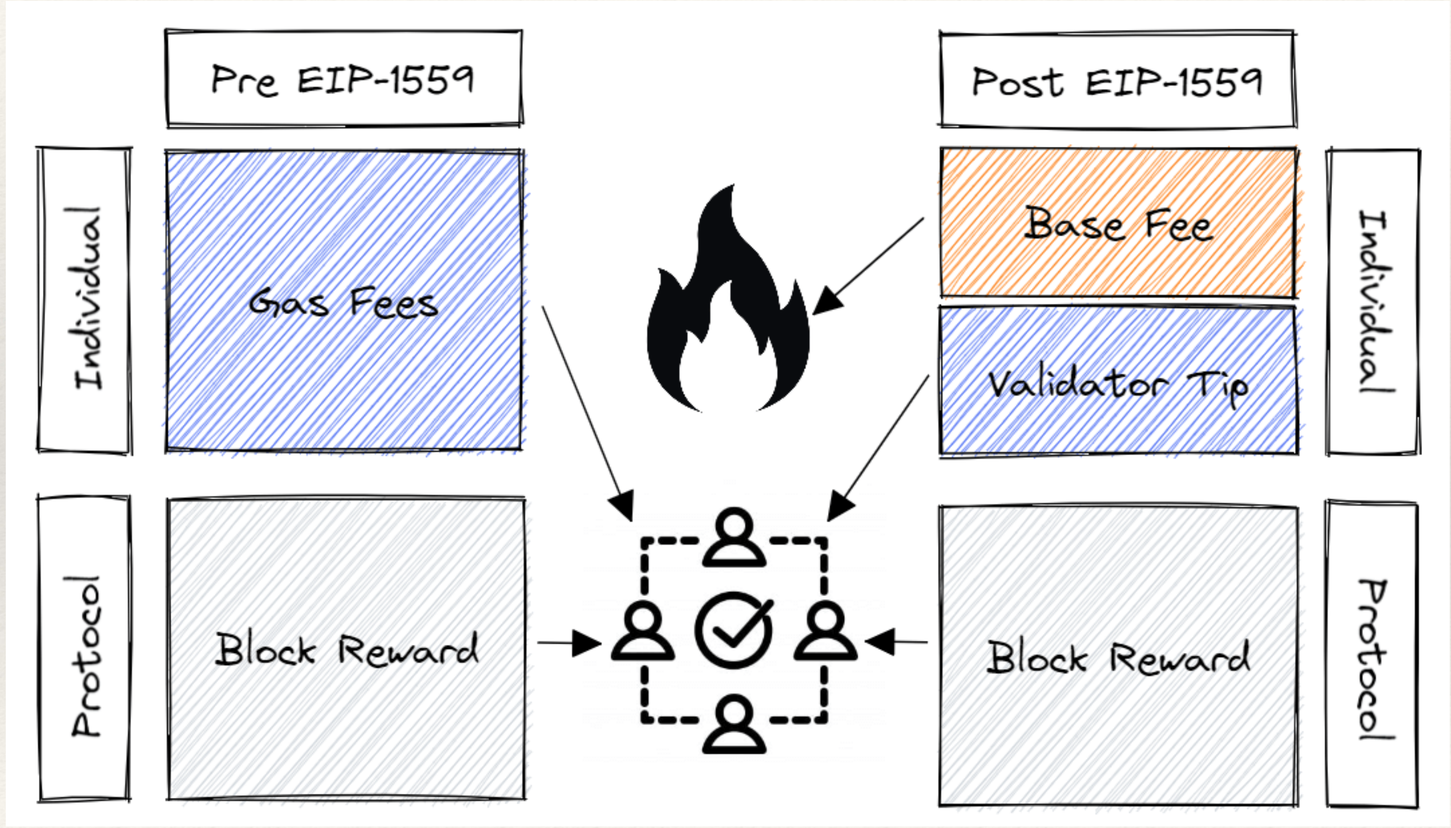
EIP1559


Other Attributes:
Txn Type: 0 (Legacy)
Nonce: 4
Position In Block: 107
Input Data:
0x


Gas Limit & Usage by Txn:
164,043 | 63,209 (38.53%)
Gas Fees:
Base: 3.089479507 Gwei | Max: 6.263313432 Gwei | Max Priority: 0.001 Gwei
Burnt & Txn Savings Fees:
Burnt: 0.000195282910157963 ETH ($0.68)
Txn Savings: 0.000200551659565325 ETH ($0.70)
Other Attributes:
Txn Type: 2 (EIP-1559)
Nonce: 339759
Position In Block: 173
Input Data:
Function: transfer(address _to, uint256 _value)
MethodID: 0xa9059cbb
[0]: 000000000000000000000000901f3d5b2d8084f4e93e6acfd317ba14a73d724f
[1]: 0000000000000000000000000000000000000000000000000000000f75bf980
View Input As
Decode Input Data
Advanced Filter
Write Contract

Gas-EIP1559

拍卖式



小费式

Gas - EIP 1559

- EIP1559 解决了 Ethereum 拥堵吗?
- EIP 1559 让用户交易体验更好了吗?
- EIP 1559 解决了什么问题?

交易Gas

- 交易时如何设置一个合约的GasLimit?

Buy: 10+11+3 - 2 = 22

22: 12, 1, out of gas

预估 API. : eth_estimateGas

GasLimit = gas * 1.1

```
import { account, publicClient } from './config'
import { wagmiAbi } from './abi'

const gas = await publicClient.estimateContractGas({
  address: '0xFBA3912Ca04dd458c843e2EE08967fC04f3579c2',
  abi: wagmiAbi,
  functionName: 'mint',
  account,
})
// 69420n
```

离线签名出售NFT

```
bytes32 private constant _LIMIT_ORDER_TYPE_HASH = keccak256(
        "LimitOrder(address maker,address nft,uint256 tokenId,address payToken,uint256 price,uint256 deadline)"
    );
```

```
function _matchMakerOrder(LimitOrder memory order⬆) private   {
    require(order⬆.deadline > block.timestamp, "MKT: order is expired");
    require(order⬆.price > 0, "MKT: price is zero");

    bytes32 orderId =_hastStruct(order⬆);

    // safe check repeat order
    require(orderMatched[orderId] == false, "MKT: order already matched");
    orderMatched[orderId] = true;

    // safe check nft owner sign
    address signer = ECDSA.recover(orderId, order⬆.signature);
    require(signer == order⬆.maker, "MKT: invalid order signature");

    //  trasnfer NFT
    IERC721(order⬆.nft).safeTransferFrom(order⬆.maker, msg.sender, order⬆.tokenId);
    // trasnfer token
    // fee 0.3% or 0
    uint256 fee = feeTo == address(0) ? 0 : order⬆.price * feeBP / 10000;
    // pay token
    _transferOut(order⬆.payToken, order⬆.maker, order⬆.price - fee);
    if (fee > 0) _transferOut(order⬆.payToken, feeTo, fee);

    emit LimitOrderMatched(msg.sender, fee, order⬆);
}
```

作业说明

- 代码在自己的 github 提交
- 在 decert.me 提交领取证书
- **不可抄袭作业**，一经发现将不再检查抄袭者作业！

作业

- 签名 NFT 上架信息
- 使用离线签名和验证存储NFT上架， 展示最新的 NFT 上架清单
- 完善 NFTMarket合约，使用ETH买入NFT
- Option：在TheGraph中记录NFT记录，并在网页中展示NFT交易动态

挑战：https://decert.me/quests/d9c6e975-4742-4f81-89fa-5986b61062ea

谢谢