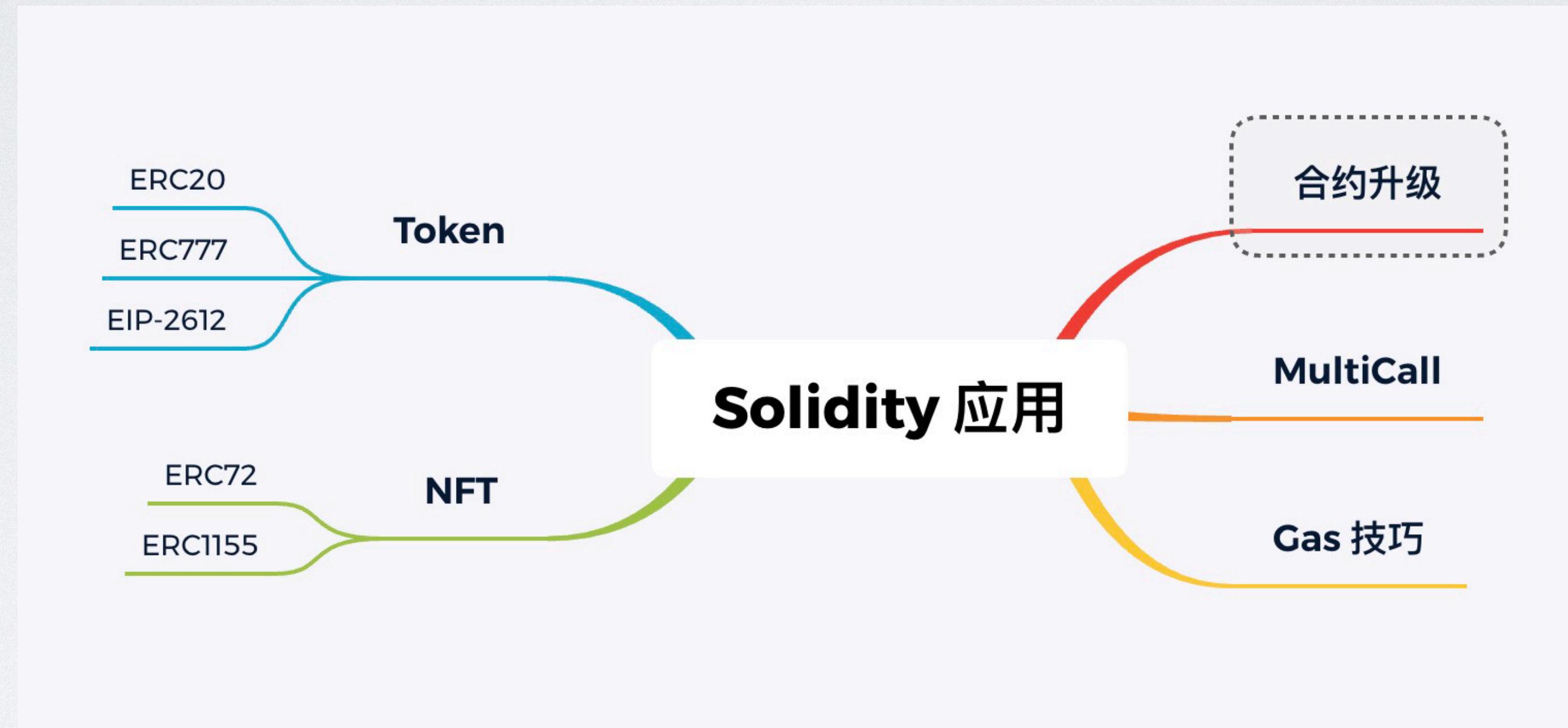


区块链集训营

二期

登链社区 - Tiny熊

W3 回顾



上一周，我们编写了 Token 合约
这周我们看看如何在前端与之交互

练习题

- 部署一个可升级的 ERC20 Token
 - 第一版本
 - 第二版本，加入方法：function transferWithCallback(address recipient, uint256 amount) external returns (bool)

习题解答

https://github.com/xilibi2003/training_camp_2

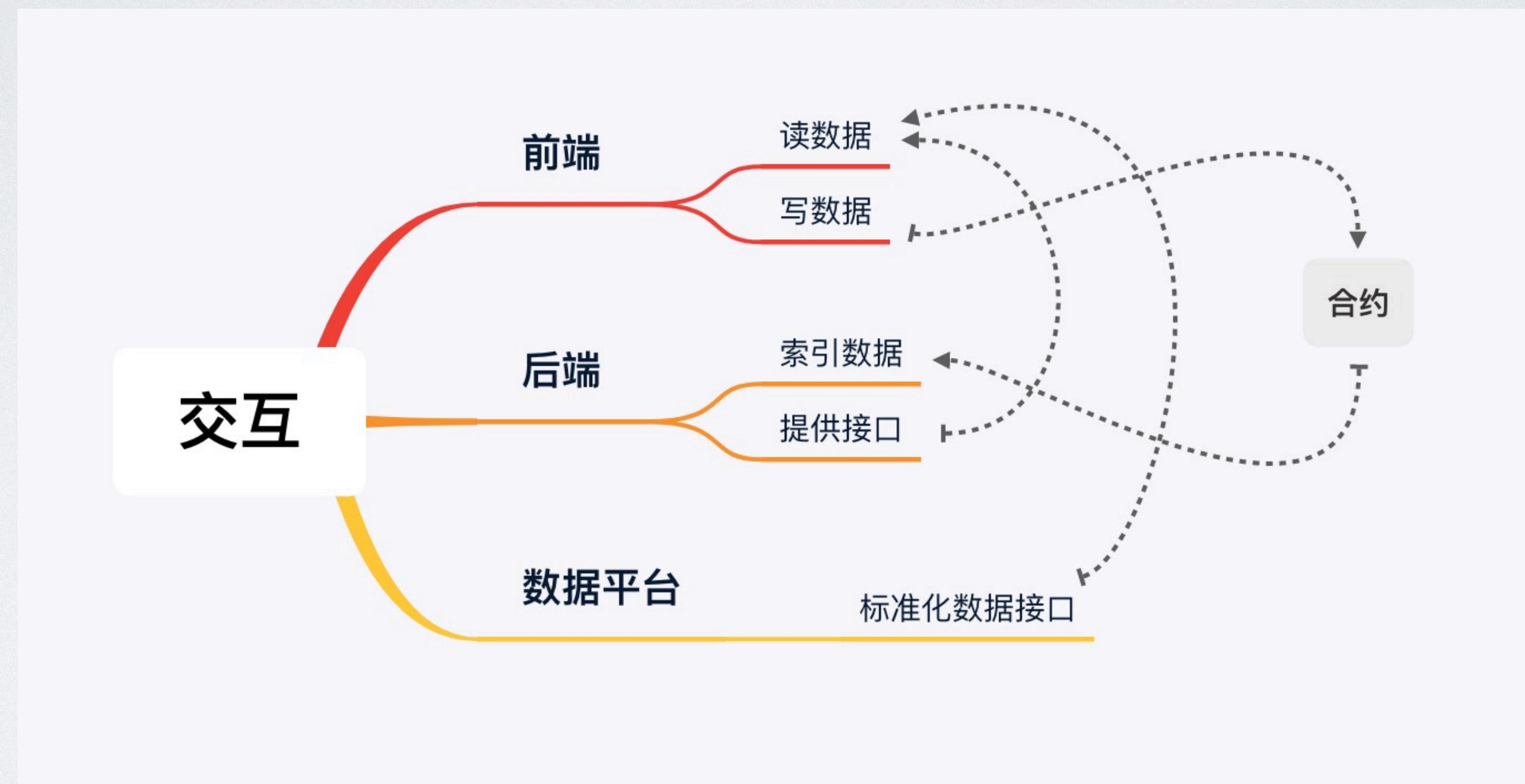
讲课代码是 main 分支
习题解答在 Answer 分支

文件：

w3_2_code/contracts/MyERC20V1.sol
w3_2_code/contracts/MyERC20V2.sol

部署地址：<https://mumbai.polygonscan.com/token/0x595c128a306ad9ae830030753e8f41201c314882>

前后端交互

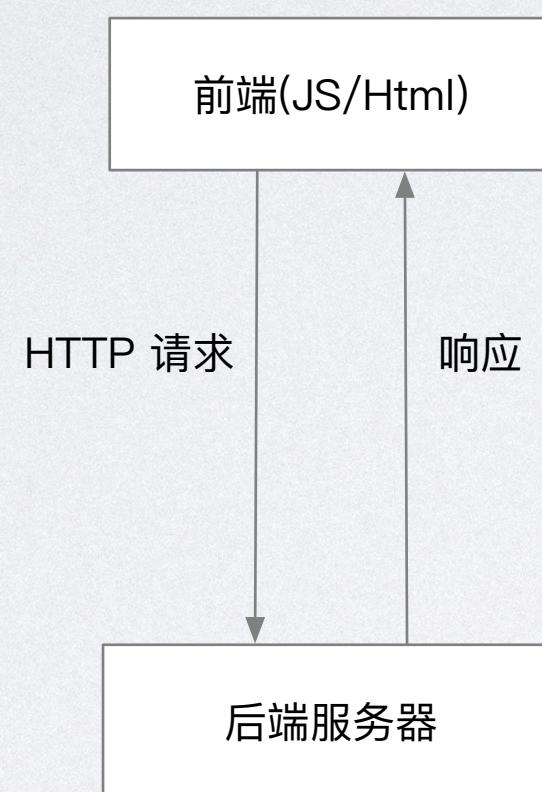


DAPP 开发实战进阶

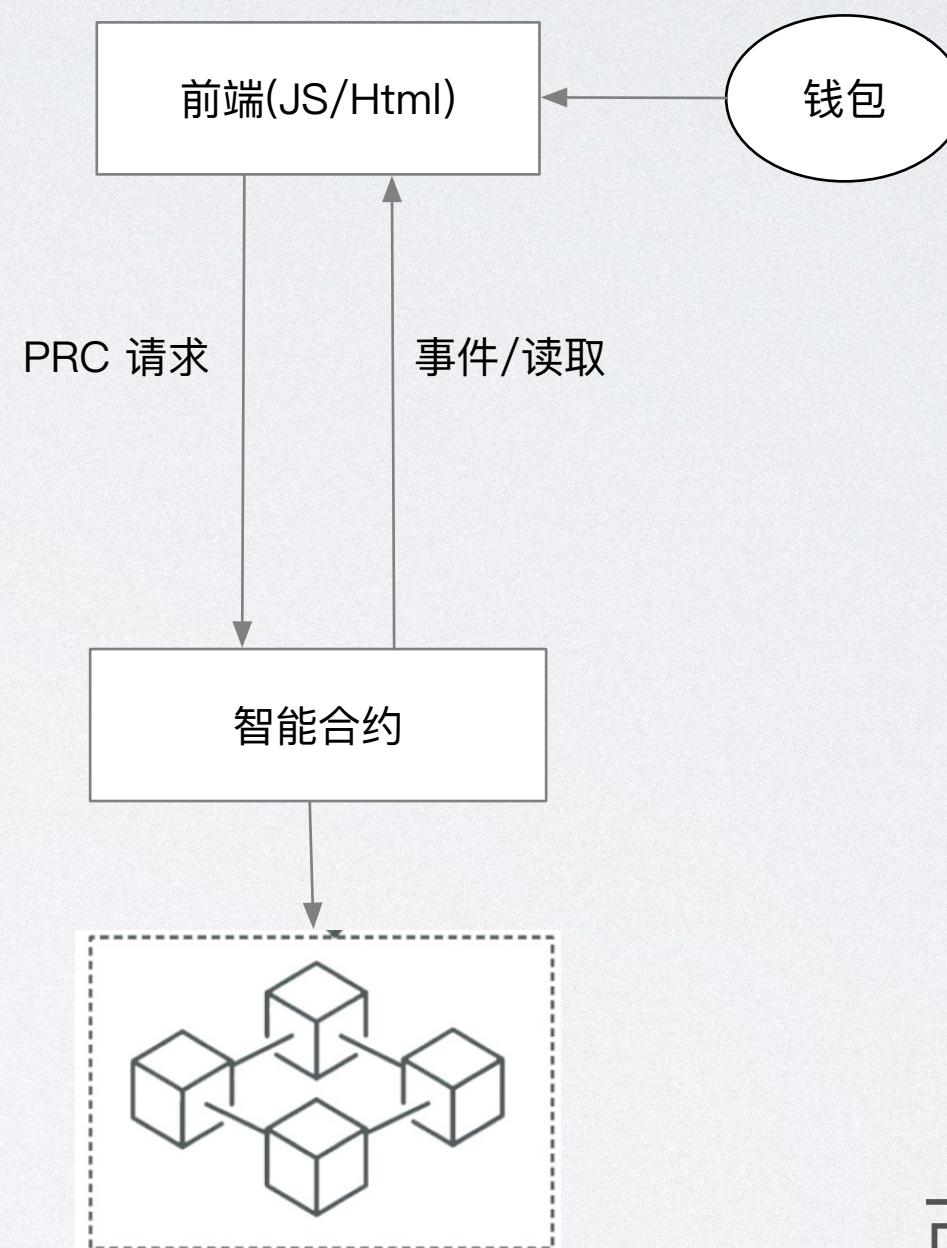
- DApp 前端开发：前端与合约交互（ethers.js）
- 后端：解析合约事件，缓存到...
- 使用第 3 方数据工具、平台
 - TheGraph 使用
 - Dune、NFT Scan、Chainbase

DApp 去中心化应用

传统应用



Web3 应用



两个主要特点：

1. 交易离不开钱包
2. 交易是异步的

DApp 去中心化应用

前端如何与合约交互

与链（节点）交互

```
// 获取账号余额  
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_getBalance","params":  
["0xe74c813e3f545122e88A72FB1dF94052F93B808f", "latest"],"id":1}' http://127.0.0.1:8545
```

```
//调用合约  
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_call","params": [{ ... }],"id":1}'
```

```
//调用合约  
curl -X POST --data '{"jsonrpc":"2.0","method":"eth_sendRawTransaction  
","params": [ { ... } ],"id":1}'
```

一个交易

调用一个合约函数 = 向合约地址发送一个交易

交易的内容就是 ABI 编
码数据

Calling a Smart Contract



Address: 0x0123456.....

1 Convert function call to HEX

myFunction(parameters) → → 0abcdef0123456789.....

2 Put the Information into Transaction object

```
{  
  "to": "0x0123456.....",  
  "value": 0, // No need to send money here  
  "data": "0abcdef0123456789....."  
}
```

3 Sign the Transaction with your Private key

{ ... } + Private Key → → 0xfedcba9876...

Signed Transaction
Can only be decrypted with YOUR public key
Only you can have sent this transaction

4 Send the transaction to the Ethereum Network

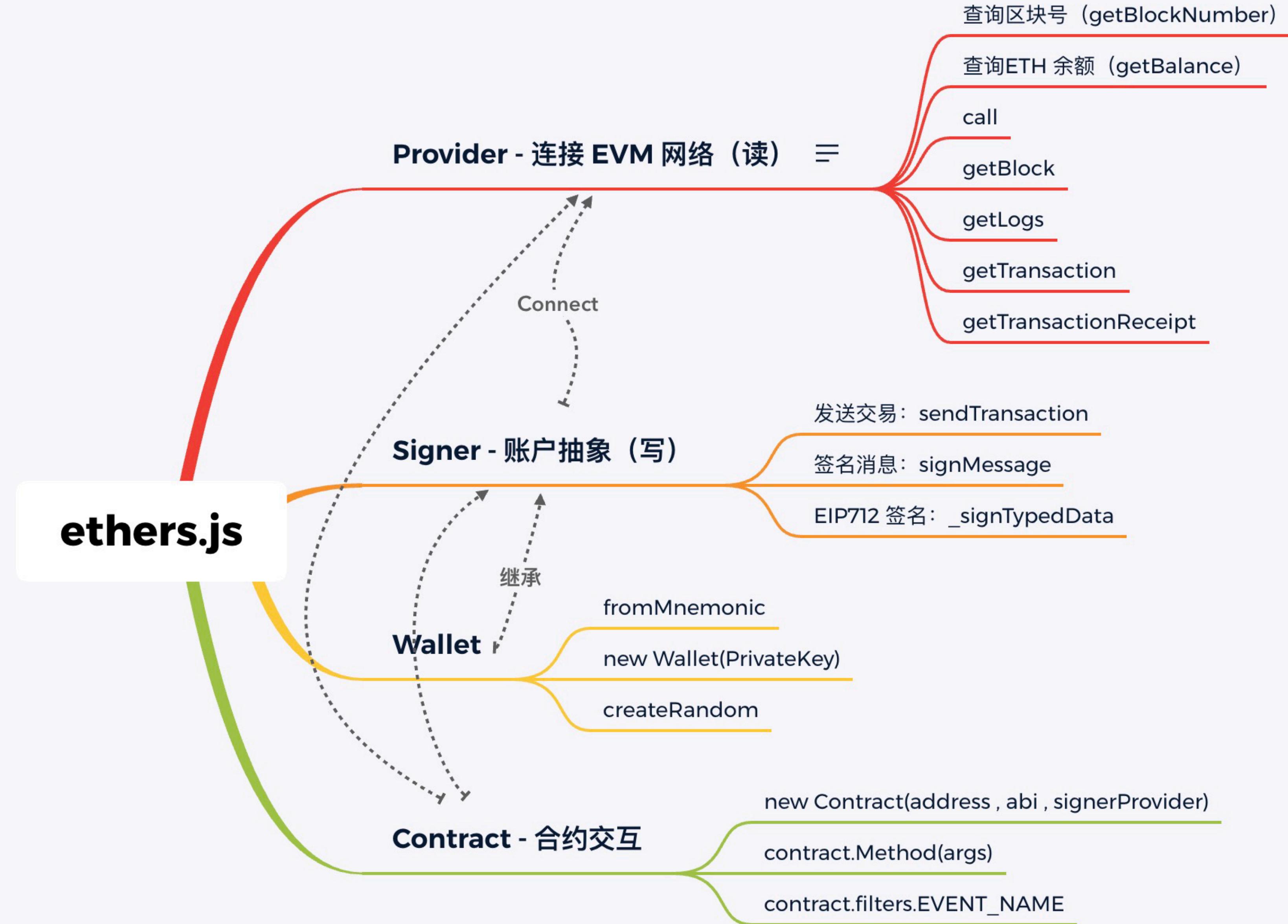


第4周

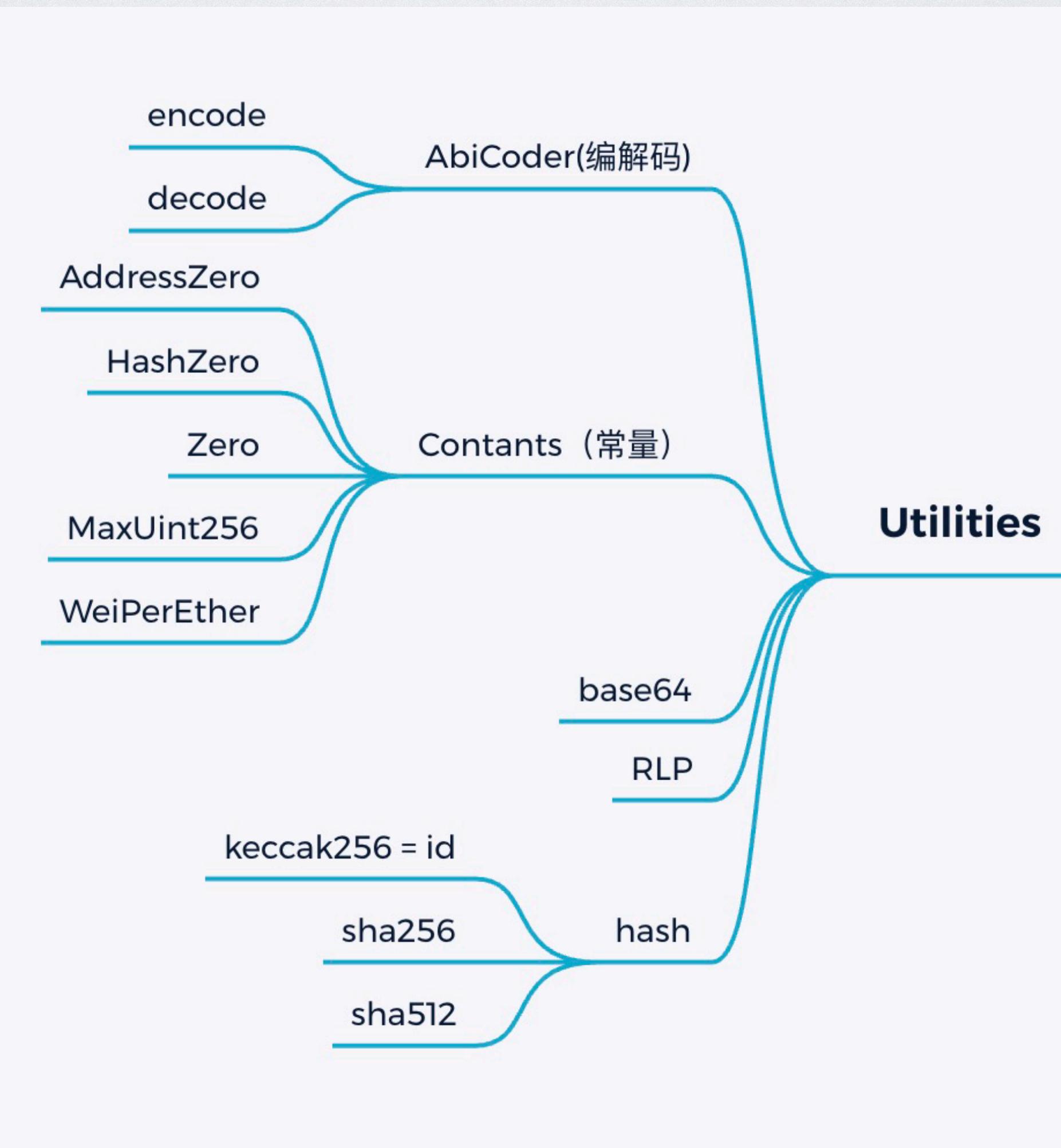
DApp 开发：前端与合约交互

- Ethers.js/Web3.js：一套和以太坊区块链进行交互的库，RPC 接口封装
- 安装：npm install --save ethers

```
await ethers.provider.getBalance("ricmoo.eth");
```



Ethers.Js



DApp 开发：前端与合约交互

- 使用 ethers.js 调用 ERC20 合约示例：

```
var abi = [...];
var addr = "0x...";
var contract = new ethers.Contract(address, abi, provider);

contract.transfer(targetAddress, amount)
  .then(function(tx) {
    console.log(tx);
  });
}
```

DApp 开发：前端与合约交互

4步：

1. 连接节点（Provider）

- MetaMask 插件会在页面中注入 `window.ethereum` 对象

2. 获取或创建钱包对象

3. 初始化合约对象(合约地址 + ABI)

4. 从合约获取数据、发起交易

MetaMask 文档：<https://docs.metamask.io/guide/>

Ethers.js 文档：https://learnblockchain.cn/ethers_v5/

DApp 开发：前端与合约交互

DEMO on Vue

https://github.com/xilibi2003/training_camp_2/tree/main/w4-vue

Q & A

练习题

- 编写前端使用之前的Vault 合约：
 - 用户可通过前端进行存款
 - 两个方式：Approve + deposit
 - 最好使用 ERC2612(Permit) 方式更好 (permitDeposit)
 - 前端显示用户存款金额
 - 用户可通过前端提取用户自己的存款 (withdraw)

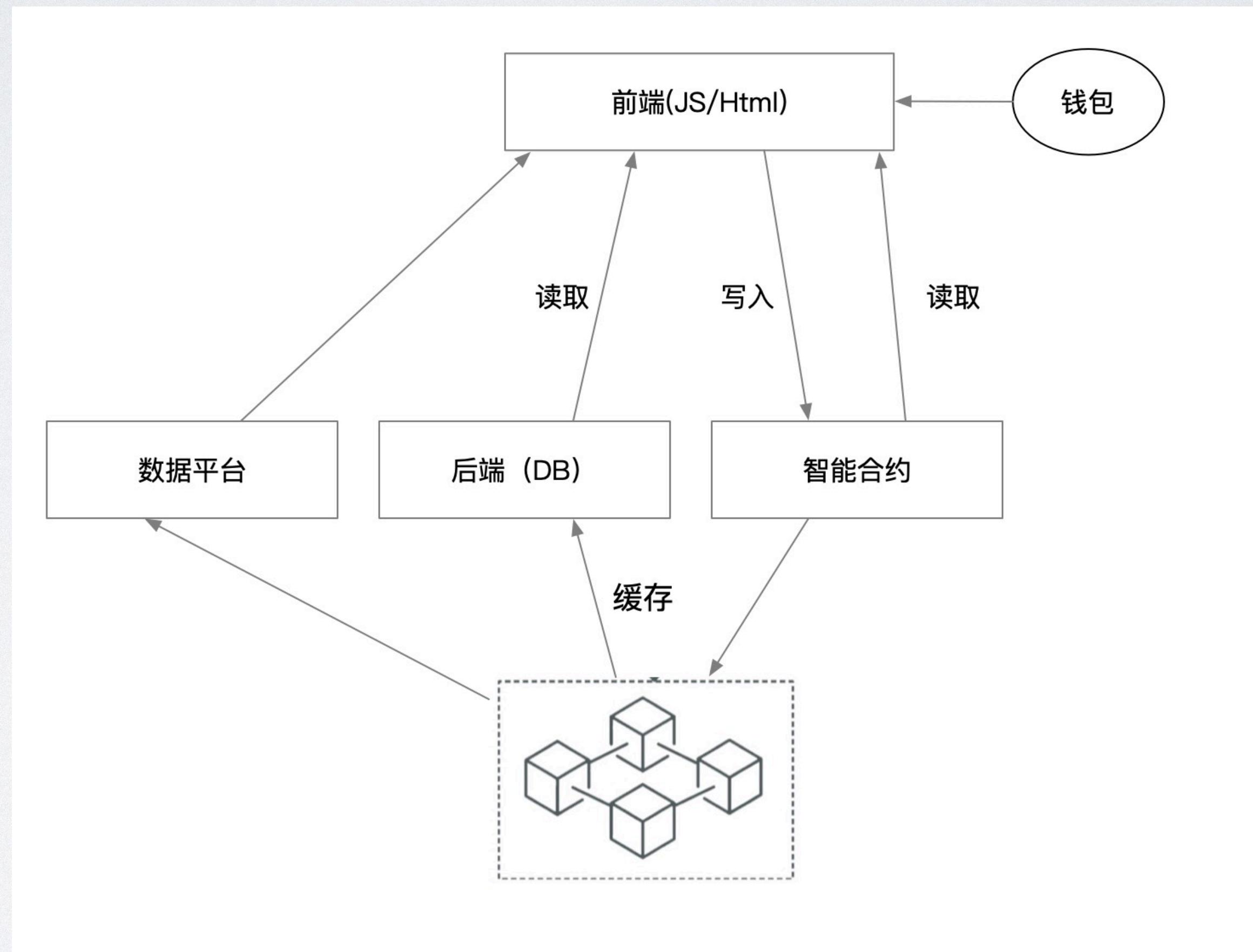
DApp 去中心化应用

后端如何与合约交互

第4周

DApp 去中心化应用

思考：如何展示某个用户的转账记录？



DAPP 应用架构

事件 - 回顾

- 我们如何知道合约状态的变化
 - 利用事件，合约在有变化时触发事件
- 事件有时也作为便宜的存储
- 使用关键字 event 定义事件，emit 触发事件
- 使用关键字 indexed 修饰，表示对这个字段建立索引，方便外部对该字段过滤查找

事件

- 事件
 - 索引的事件 生成 对应的 Topic
 - 每个事件有个事件签名作为Topic
 - 非索引事件作为 data 保存

```
let transferTopic = ethers.utils.keccak256(  
    ethers.utils.toUtf8Bytes("Transfer(address,address,uint256)"));  
  
// 等价  
ethers.utils.id("Transfer(address,address,uint256)")
```

https://learnblockchain.cn/ethers_v5/api/utils/hashing/

监听事件

- 监听（实时）事件（使用 webSocket 链接）

```
let erc20 = new ethers.Contract(erc20地址, ABI, provider);

// indexed 的字段可以作为参数加入到过滤器中
let filter = erc20.filters.Transfer()

wssprovider.on(filter, (log) => {
    console.log(log);
})
```


获取事件

- 获取（历史）事件

```
let erc20 = new ethers.Contract(erc20地址, ABI, provider);
let filter = erc20.filters.Transfer()

// 选择区块区间
filter.fromBlock = 1000;
filter.toBlock = 2000; // "latest";

// 获取日志
let logs = await provider.getLogs(filter);
let logs = await erc20.queryFilter(filterTo, -10, "latest");
```

https://learnblockchain.cn/ethers_v5/api/providers/provider/#Provider--log-methods

处理事件

- 解析事件
 - 使用 ethers.utils.Interface 进行编解码

```
const TransferEvent = new ethers.utils.Interface(["event Transfer(address indexed from,address indexed to,uint256 value)"]);

const data = TransferEvent.parseLog(logs[i])
console.log("from:" + data.args.from)
console.log("from:" + data.args.to)
```

https://learnblockchain.cn/ethers_v5/api/utils/abi/interface/

DApp 开发：后端索引合约事件

DEMO with Node.js

https://github.com/xilibi2003/training_camp_2/tree/main/w4-backend

Q & A

练习题

- 发行一个 ERC721 Token：
 - 使用 ethers.js 解析ERC721 转账事件
 - 加分项：记录到数据库中，可方便查询用户持有的所有 NFT