

DID

1. Enumerating Contracts Events to build the DID Document (2 Events)

1. `DIDOwnerChanged` (indicating a change of `controller`)
2. `DIDAttributeChanged`

```
event DIDAttributeChanged(  
    address indexed identity,  
    bytes32 name, //由于gas效率的原因不使用string  
    bytes value,  
    uint validTo, //有效期时间  
    uint previousChange //记录上一个属性改变的区块号  
);
```

2. 方便查询事件

1. 策略

1. 合约维护一个名为 `changed` 的映射，记录最新的改变所在的区块号
2. 每个Event都有 `previousChanged` 的属性，用来存储上一个更改事件的块号

2. 查询Event所有历史的步骤

1. `eth_call changed(address identity)` on the contract to get the latest block where a change occurred
2. If result is `null` return.
3. Filter for events for the above 3 types with the contract address on the specified block
4. If event has a previous change then go to 3

3. 虽然可以存储任何属性，但对于 DID 文档，我们支持添加到 DID 文档的以下每个部分：

- `verificationMethod`
- `authentication`
- `proxywallet`
- `service`

4. Attribute Name 遵循以下命名规则，以满足DID Document属性的层次要求

- `verificationMethod`
 - `vfm/id: id`
 - `vfm/type: type`
 - `EcdsaSecp256k1VerificationKey`
 - `Ed25519VerificationKey2020`
 - `vfm/con: controller`
 - `vfm/pkm: publicKeyMultibase`
 - `vfm/eth: ethereumAddress`
- `authentication`
 - `atc/vfm/id: verificationMethod id`
 - `atc/id: id`

- `atc/type`: type
- `atc/con`: controller
- `atc/pkm`: publicKeyMultibase
- proxywallet
 - `pxw/vfm/id`: verificationMethod id
 - `pxw/id`: id
 - `pxw/type`: type
 - `pxw/con`: controller
 - `pxw/eth`: ethereumAddress
- service
 - `svc/id`: id
 - `svc/type`: type
 - Linked Domains
 - DIDCommMessaging
 - CredentialRegistry
 - `svc/ep`: endpoint

6. 权限管理

1. 创建不需要权限需要指定owner（一般钱包地址）
2. 更改/新增属性需要权限，owner来调用合约

7. 记录创建时间

- Attribute Name: created

DID合约部署

合约代码:

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.6;

contract EtdDIDReg{

    bytes32 constant private KEY_CREATED = "created";

    mapping(address => address) public owners;
    mapping(address => uint) public changed;

    modifier onlyOwner(address identity, address actor) {
        require (actor == identityOwner(identity), "bad_actor");
        _;
    }

    event AttributeChanged(
        address indexed identity,
        bytes32 name,
        bytes value,
        uint previousBlock,
        uint validTo
    );
}
```

```

event OwnerChanged(
    address indexed identity,
    address owner,
    uint previousBlock
);

function createId(
    address identity,
    address controller,
    uint validTo
)
    public
{
    uint currentTime = block.timestamp;
    bytes memory creationTime = abi.encodePacked(currentTime);
    owners[identity] = controller;
    changed[identity] = block.number;
    emit OwnerChanged(identity, controller, 0);
    emit AttributeChanged(identity, KEY_CREATED, creationTime, 0, validTo);
}

function changeOwner(address identity, address newOwner) public
onlyOwner(identity, msg.sender) {
    owners[identity] = newOwner;
    changed[identity] = block.number;
    emit OwnerChanged(identity, newOwner, changed[identity]);
}

function setAttribute(address identity, bytes32 name, bytes memory value,
uint validTo ) public onlyOwner(identity, msg.sender) {
    emit AttributeChanged(identity, name, value, changed[identity],
validTo);
    changed[identity] = block.number;
}

function identityOwner(address identity) public view returns(address) {
    address owner = owners[identity];
    if (owner != address(0x00)) {
        return owner;
    }
    return identity;
}
}

```

ETD合约地址: 0x85858fe01A9EF40f956B259c5709942Ffa074F5B

[illegible]


```
[{"value": "0x0000000000000000000000000000000000000000000000000000000000000064d111f6",  
  "previousBlock": "0",  
  "validTo": "0"  
}]
```

- `setAttribute`
 - `input`

```
decoded input {
  "address_identity": "0x1cc5dc930549740508b253cd19a37f9b48eD79b7",
  "bytes32 name": "0x76666d2f69640000000000000000000000000000000000000000000000000000",
  "bytes value": "0x2370726f787977616c6c65742d31",
  "uint256 validTo": "0"
}
```

```
//blocknumber
5991909
//log
[
  {
    "from": "0x3a1bb4f1826A555CC5B0b20B61e1137E178dc4Cc",
    "topic":
      "0x024a588a8e0cc47a69d57e77c56bc4525491dafb13d7d2553025177d4aeb9746"
    ,
    "event": "AttributeChanged",
    "args": {
      "0": "0x1cc5dc930549740508b253cd19a37f9b48ed79b7",
      "1":
        "0x7666d2f696400000000000000000000000000000000000000000000000000000"
      ,
      "2": "0x2370726f787977616c6c65742d31",
      "3": "5991880",
      "4": "0",
      "identity":
        "0x1cc5dc930549740508b253cd19a37f9b48ed79b7",
      "name":
        "0x7666d2f696400000000000000000000000000000000000000000000000000000"
      ,
      "value": "0x2370726f787977616c6c65742d31",
      "previousBlock": "5991880",
      "validTo": "0"
    }
  }
]
```

- changed

```
decoded output      {
                      "0": "uint256: 5991909"
                      } 
```

- 封装DID Document
 - metadata

- event OwnerChanged
- event AttributeChanged

☐ 创建DID API

- 提供创建数据
- 输出封装DID Document

☐ 更改DID Document Attribute API

- 提供属性name & value
- 输出封装DID Document

DNS

☐ DNS合约完善