Meta42 权限数据链上存储、溯源的 Violas 区块链解决方案

版本	作者	修改时间	说明
0.1	孙海涛	2022/4/8	初稿供讨论
0.2	孙海涛	2022/4/13	加入智能合约的实现代码和详细说明

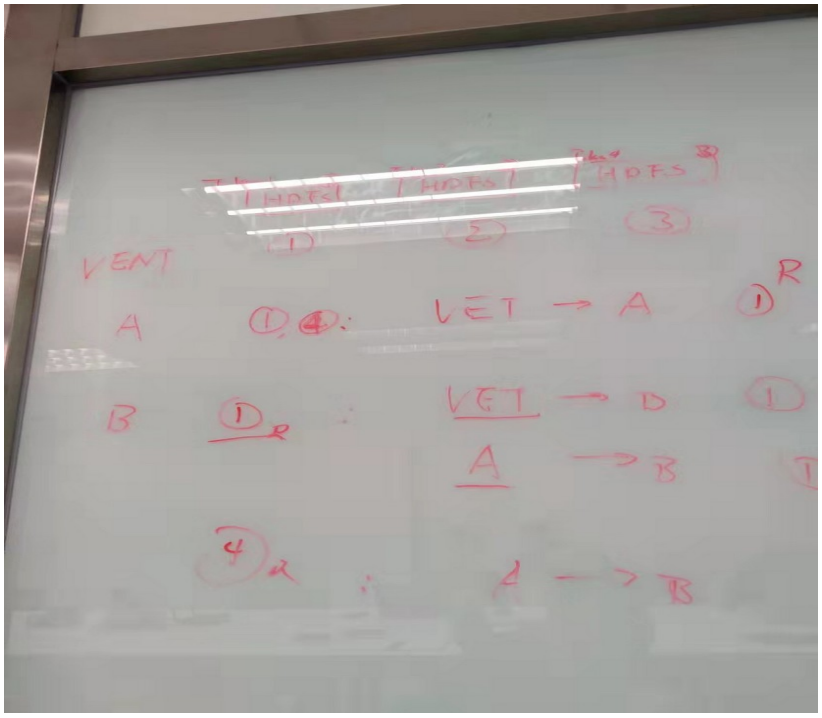
1 项目背景介绍

Meta42 项目首先立足于数据中心 42U 标准机柜的全面感知、智能运维和数字孪生，进一步打造拥有独立算力芯片、开源开放体系的数据中心新一代机柜。Meta 42 项目的长期愿景是工业元宇宙的实验和探索，专注面向全球 ET+IT 混合设备创造元宇宙独数字资产，以及数字资产运营和交易。Meta 42 是个工程师社群主导的去中心化科研探索项目，包括三个工作组：开源软件工作组、硬件和芯片工作组、去中心化运维 DAO 工作组。

2 需求概述

Meta42 项目需要将链外数据资产的访问权限记录到区块链上。利用 Violas 区块链的不可篡改、链上交易可追溯性，能够对权限数据(Token)进行创建、分享、溯源，同时根据链上权限数据（Token）的所有权在链外对访问数字资产的用户进行鉴权后授权访问。

2.1 原始需求图例



在会议讨论的原始图例中， HDFS 中有三个数据块，VNET 在链上创建三个对应的 Token1、Token2 和 Token3，用户 A 创建另外一个 HDFS4 同时在链上创建了 Token4.

期待实现的链上功能是

1. VNET 将 Token1 分享给用户 A
2. 用户 A 将 Token1 分享给用户 B
3. 用户 A 将 Token4 分享给用户 B

在当前状态下，需要通过 Violas 区块链判断某个 Token 的拥有权，还需要对 Token 的分享历史能够溯源。

下面将会议的原始需求做了一个初步的需求分析。

2.2 文档中使用到的术语说明

- **Token**

  链上的一个 Token 代表链外的一个数字资产的详细描述，可有很多子项例如 http url, HDSF 路径等...

  Token {

  hdfs_path : string,

  xxx

  }， Token 中可以保存多个 xxx 子项，目前智能合约中 Token 只有 hdfs_path，保存链外的 HDFS 文件路径。

- **Token 所有权**

  Token 存储在哪一个账户下面，就代表这个账户对 Token 的所有权。同时，这个账户根据 Token 中子项的描述信息有权限访问链外的数字资产。

- **Token Id**

  Token Id 代表的 Token 的唯一标识, 具有唯一性。

  Token Id 的计算方式:

  1. BCS 算法(Violas SDK 提供）序列化 Token 中的所有数据项，得到字节流
  2. 对步骤 1 生成的字节流使用 sha3-256 哈希算法计算 hash ，产生 32 个字节。

- **Meta42 账户**

  由 Meta42 智能合约管理员调用 create_child_vasp_account 创建的子账户，才能够调用 Meta42 合约的接口。

  Violas 链上的其它账户无权限调用 Meta42 智能合约接口。

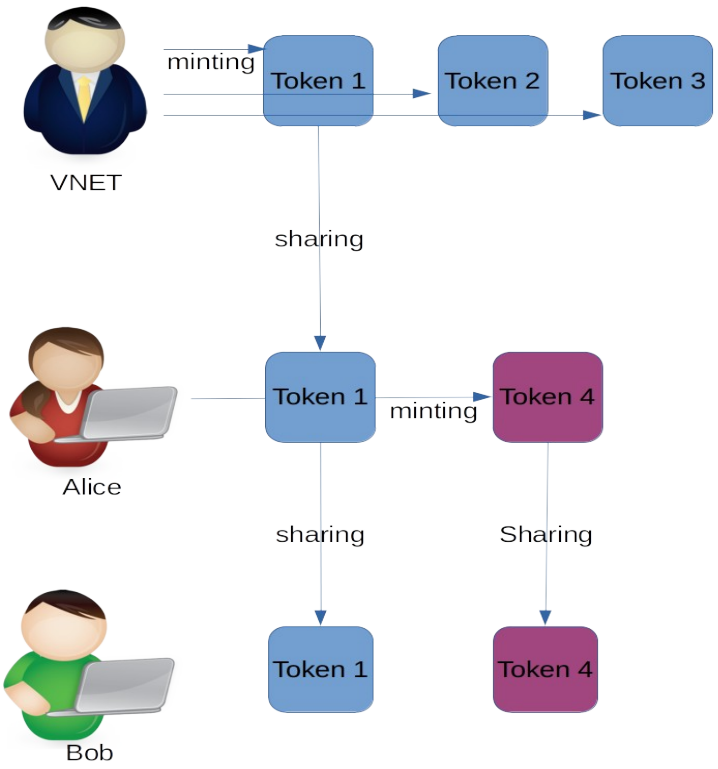
2.3 功能需求列表

功能需求	说明
1. 创建 Token	在 Violas 链上创建一个 Token（链外资产的描述结构）, 此 Token 的所有权为当前创建账户。
2. 分享 Token	将某个 Token 的分享(拷贝)给另外一个账户，Token 的能够被多个账户拥有。
3. 溯源 Token	根据一个 Token Id, 追溯对应的 Token 被分享的所有历史信息
4. 链外鉴权	当一个用户去访问链外的数据资产时，链外后台服务需要能够判断当前的用户的链上身份，以及是否有权限访问链外的数据资产。账户身份和链外资产对应 Token 的拥有权在 Violas 链上都可以查询。
5. 创建子账户	Violas 链原生提供创建子账户的功能

3 用于验证的测试用例

3.1 测试步骤

1. Violas 管理员创建 VNET 账户， VNET 属于 Parent VASP 角色。
2. VNET 账户创建子账户 Alice 和 Bob， Alice 和 Bob 属于 Child VASP 角色。
3. VNET 账户铸造(mint) Token 1, Token 2, Token 3
4. VNET 账户分享(share) Token 1 给账户 Alice
5. Alice 铸造(mint) Token 4
6. Alice 分享(share) Token1 和 Token 4 给账户 Bob
7. 测试步骤的状态图如下:



3.2 Tokens 的拥有权的期待结果

账户	角色	Tokens			
VNET	Parent VASP	Token 1	Token 2	Token 3	
Alice	Child VASP	Token 1			Token 4

Bob	Child VASP	Token 1			Token 4

3.3 溯源 Token 的期待结果

期待的结果如下

- 对 Alice 的 Token 1 溯源: VNET→ Alice
- 对 Bob 的 Token 1 溯源: VNET → Alice → Bob
- 对 Bob 的 Token 4 溯源 : Alice → Bob

4 Meta42 使用的 Move 智能合约

4.1 智能合约源码

https://github.com/violas-core/violas-client-sdk/tree/main/move/meta42

4.2 编译 Meta42 智能合约

Ubuntu 20.04 环境，已安装 Move 语言编译器 move, 执行如下命令

```
move compile meta42.move scripts.move  --mode diem
```

在 build 目录生成编译后的 meta42.mv 和多个脚本 meta42*.mv

4.3 智能合约的接口

使用 Violas SDK 和编译出的 meta42*.mv 脚本字节码，即可以调用如下的接口，详细的链上交易提交方法请参考官方 SDK 文档。

接口名称	功能	参数说明
initialize	初始化 Meta42 合约	调用权限：Meta42 管理员账户
accept	注册用户信息	调用权限：任何 Meta42 普通账户, 说明 : 1. 一个账户在必须调用 accept 接口之后，才能接收别人的分享 Token, 只需调用一次。 2. mint_token 会在内部调用 accept, 如果用户没有调用 accept。
mint_token	铸造一个 token 到自己的账户	参数: **hdfs_path** – 链外的 HDFS 文件的路径, 将会保存在 Token 中。 权限: 任何 Meta42 用户
share_token_by_index	分享 token 给另一	参数:

	个账户	**receiver** – 接收者账户地址 **index** – token 在当前账户下的索引，根据客户端能够获取所有的 token, 索引从 0 开始到 n-1 . **Message** – 附加的信息 权限：任何 Meta42 账户
share_token_by_token	分享 token 给另一个账户	参数: **receiver** – 接收者账户地址 **token_id** – token 的唯一标识. **Message** – 附加的信息 权限：任何 Meta42 账户
create_child_vasp_account	创建子账户	参见 Violas SDK 接口说明。

4.4 链外的功能（客户端实现）

4.4.1 获取账户地址的所有 tokens

客户端使用 SDK 通过 Violas Json API 获取到某个账户地址下的所有 tokens.

4.4.2 获取某个账户铸造（mint)历史记录

客户端使用 SDK 中的 get_event API 获取账户下 Account Info 中的 MintedTokenEvent.

4.4.3 获取某个账户中的自己分享 Token 给别人的历史记录

客户端使用 SDK 中的 get_event API 获取账户下 Account Info 中的 SentTokenEvent.

4.4.4 获取某个账户中别人分享 Token 给自己的历史记录

客户端使用 SDK 中的 get_event API 获取账户下 Account Info 中的 ReceivedTokenEvent.

5 后台服务功能

5.1 Token 溯源

Violas 区块链合约中将会使用一个 SharedTokenEvent 结构，如下

```
SharedTokenEvent {
        token_id : bytes,
        sender: address,        // 分享 token 的发送者
        receiver: address,      // 分享 token 的接收者
        message : vector<u8>
```

};

这个结构体在 Meta42 智能合约的管理员账户下，存储了链上的所有分享 Token 的历史信息，单向增长且永不改变，一直保存在 Violas 区块链上。这个结构可以按照索引序号查询，但是缺少 SQL 中的聚合查询功能。如果将所有的 SharedTokenEvent 信息同步到本地数据中，就可以使用 SQL 可以进行灵活方便的查询。

推荐的实现方式：同步(get_event Json API)链上合约中的 SharedTokenEvent **到本地数据中，使用 SQL 查询某个 Token id 的的分享历史信息**。

5.2 链外判断 Token 拥有权

这里讨论的是一个可行的方法，仅供参考。假设有一个后台服务，如果需要判断一个用户是否在 Violas 区块链上拥有某个 Token, 用户需要使用 Violas 的 ED25519 的签名机制，对 token Id 做一个签名后提交到后台服务，后台服务使用是个带签名的消息查询 Violas 区块链就可以验证用户的身份和检查是否拥有对应的 Token 。

```
Struct AuthenticationMessage {
        token_id : byte32;
        signature : Signature; // 对 Token ID 的 ED25519 签名
        Public Key;    // 用于验证签名的公钥
        address:  Address;    // 链上的账户地址，使用上面的公钥也可以推导出正确的地址
};
```

后台服务的判断过程如下:

1. 使用 Message 中的公钥验证签名，证明签名是有效的。
2. 使用公钥推导出一个账户地址，并且和 Address 比对，如果相等证明签名的人拥有这个账户地址。
3. 到 Violas 区块链上查询 address 拥有的所有 Token，如果查询到对应的 Token ID, 判断出 address 的账户拥有相应的 Token。