DeFi结合NFT-NFT拆分售卖的智能合约开发和测试

NFT可以代表链上独一无二的资产，比如加密猫，decentraland中的土地等等。但基于NFT的定义和技术规范ERC721,ERC1155，用户不能购买一个NFT的一部分，即NFT不能拆分，要么拥有整个NFT，要么不拥有。

今天演示一下结合DeFi的方式，实现购买一个NFT拆分后的一部分，这样的话多个用户就可以共同拥有一个NFT，比如共同拥有一幅画。

具体实现方式就是实现一个Re-Fungible Token，简称RFT，允许用户只购买一个NFT的一部分。

整体流程

* 第一步，一个用户（Admin）在交易所比如OpenSea上通过Ether或DAI购买了一个NFT，这是独一无二的，比如一幅画。用户付出了Ether或DAI，得到了一个NFT。NFT的所有权从卖家转移到了购买者。

文本

描述已自动生成

* 第二步，用户(Admin)创建并部署一个Re-Fungible token ERC20智能合约，并把自己拥有的NFT发送到这个智能合约，即NFT的所有权从用户转移到了Re-Fungible token合约。

手机屏幕截图

中度可信度描述已自动生成

* 第三步，用户（Admin）就可以通过这个Re-Fungible Token合约出售这个NFT的一部分，无数买家都可以参与购买一定的份额。买家付出DAI，获取NFT的一部分份额。

手机屏幕的截图

中度可信度描述已自动生成

售卖结束之后，Admin就可以取出Re-Fungible Token合约中的DAI，以及剩余没有卖出的NFT份额。

* 第四步，拥有RFT token的用户（即已经买了NFT的一部分份额的用户）就可以通过去中心化交易所，比如Uniswap，出售他们的RFT token。

具体实施---开发一个Re-Fungible Token智能合约

首先安装nodejs和truffle，由于我早已经安装过了，就不演示了。

新建一个目录re-fungible-tokens，初始化一个truffle项目，执行npx truffle init

文本

描述已自动生成

truffle-config.js中solidity的版本是0.8.9

初始化npm项目，执行npm init -y

文本

描述已自动生成

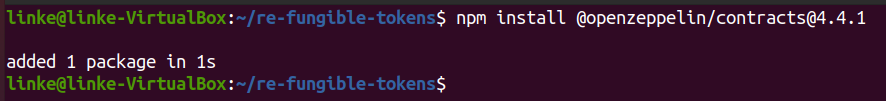
可以看到，多了一个package.json文件。

接下来安装openzeppelin，选择最新的版本4.4.1

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

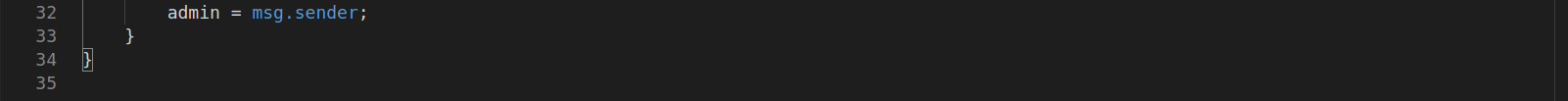
执行npm install @openzeppelin/contracts@4.4.1



在contracts目录下创建智能合约FungibleNFT.sol，先写构造函数

文本

描述已自动生成



执行npx truffle compile，编译一下

文本

描述已自动生成

接下来定义一个方法来启动RFT的I~C~O

文本

描述已自动生成

必须admin（即NFT的拥有者）才能启动I~C~O，admin需要把自己在NFT合约中的一个NFT token转移到RFT合约，即所有权转移，最后设置一下I~C~O的结束时间为一个星期后。

接下来再定义一个购买NFT份额的方法

电脑屏幕截图

描述已自动生成

首先确保I~C~O已经启动以及还没有结束，然后再判断RFT tokens是否已经卖完，每次买RFT tokens的时候，就会为买家铸造一定数量的RFT tokens。然后把买家的DAI从DAI合约转移到RFT合约，最后再为买家铸造相应数量的RFT tokens。

接下来再写一个方法，用于admin（即NFT份额卖家和I~C~O发起人）取出RFT合约中的DAI，以及剩余的没有卖出的RFT tokens。

文本

描述已自动生成

执行npx truffle compile，编译

文本

低可信度描述已自动生成

测试RFT智能合约

在truffle工程下，有一个test目录

文本

描述已自动生成

自动化测试脚本就可以写在里面。

但在测试之前，需要先模拟开发一个NFT合约，以及一个DAI合约，因为RFT合约要和这两个合约交互。

创建一个NFT合约，如下

文本

描述已自动生成

之所以定义一个mint方法，主要是为了测试，测试前首先得给admin铸造一个NFT，供其拿来售卖。

DAI合约模拟如下

电脑萤幕的截图

描述已自动生成

再执行npx truffle compile

电脑萤幕画面

描述已自动生成

至此，所有的合约都已经准备好了。接下来安装测试依赖，@openzeppelin/test-helpers

npm install @openzeppelin/test-helpers

图片包含 图形用户界面

描述已自动生成

接下来创建并开发测试脚本test/RFT.js，内容如下

文本

中度可信度描述已自动生成

测试脚本通过代码测试了NFT的创建，转移到RFT合约进行售卖，4个买家购买总共100000份，模拟I~C~O结束后卖家取出4个买家支付的DAI的整个过程。具体细节请看上图代码，就不解释了，代码已经比较清晰。

接下来运行测试

npx truffle test

报错说：Cannot find module 'web3'

因为truffle是全局命令，依赖全局web3，所以需要全局安装web3

sudo npm install -g web3

注：如果网络有问题，需要用npm config set proxy=[http://\*\*\*\*:\*\*\*\*](http://****:****)设置代理。

文本

描述已自动生成

再次执行npx truffle test，测试成功

文本

描述已自动生成

注意到测试时使用的test network，这是truffle test启动的一个本地内存区块链网络，测试完之后就不存在了，专门用于测试。

至此，就完成了把一个NFT拆分成100000份进行售卖，并用测试脚本测试智能合约的整个过程。接下来就应该把智能合约部署到ropsten或kovan等测试网上，这里就不演示了。