

# NFT 盲盒 + 市场

Daniel @ NEAR



# 盲盒





# NFT 盲盒

- 事先创作好一系列的NFT
- 买家事先并不知道会抽中哪一个
- 在购买的时候随机抽选一个

# NFT盲盒常见问题

- Metadata的生成、存储问题
- 随机算法

# Metadata

- 如何解决随机抽取NFT盲盒后，Metadata动态确定的问题
- 链上：将所有NFT的metadata信息事先存入链上，由抽取方法负责读取和即时生成
- 链下：将所有NFT的metadata信息存入链下，并用可预测的Token ID建立索引。

# Metadata

```
NFTContract {  
    metadata: [{..}, {..}, {..}, ..]  
  
    fn draw() {  
        token = new Token()  
        token.id = randomId()  
        token.metadata = metadata[token.id]  
        return token  
    }  
}
```

# Metadata

## 链上的优点

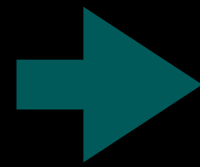
1. 实现容易
2. 不依赖于第三方系统（去中心化 + 消费者友好）

## 链上的缺点

1. 无法有效存储大多数多媒体文件
2. 存储费用高

# Metadata

ID 1: xxx  
ID 2: xxx  
ID 3: xxx  
...



```
NFTContract {  
  baseUrl = 'https://s3.aws.com/'  
  
  fn draw() {  
    token = new Token()  
    token.id = randomId()  
    token.metadata_url = baseUrl + token.id  
    return token  
  }  
}
```



# Metadata

## 链下的优点

1. 存储成本低
2. 可以利用CDN等技术加速媒体文件传输

## 链下的缺点

1. 额外搭建、配置成本
2. 第三方服务的可靠性
3. metadata可修改

# IPFS ?



# IPFS

# 随机抽取ID

```
NFTContract {  
  
    ids: []  
  
    fn randomId() {  
        while (true) {  
            id = rand()  
            if (id not in ids) {  
                ids.push(id)  
                return id  
            }  
        }  
    }  
}
```

# Fisher Yates

