**溯源原型系统**

**（设计文档）**

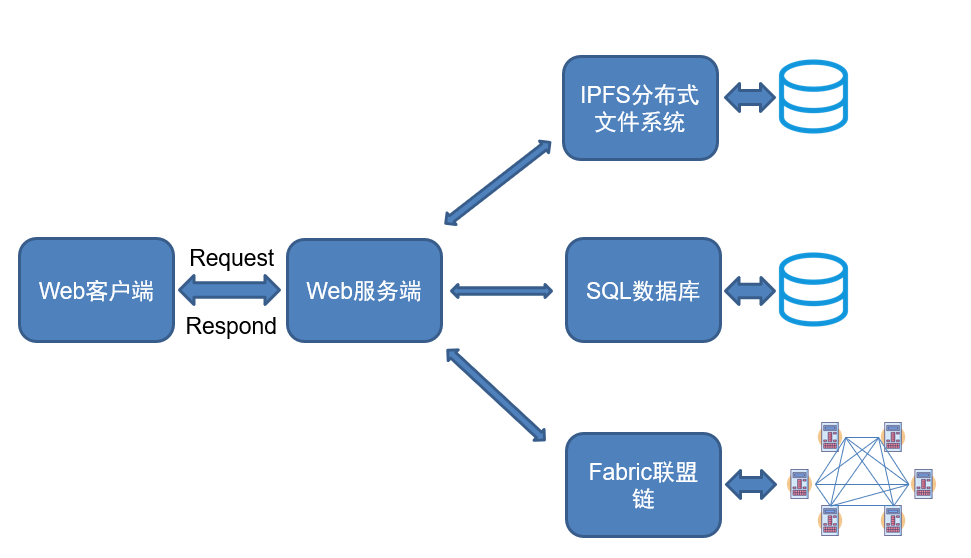
### 1. 功能概述

* 基于区块链技术对当前的数据集存储机制进行优化。利用区块链高冗余、防篡改、分布式的数据存储特性维护数据集元数据，提高数据集存储的公信度；基于区块链技术在用户间建立分布式信任，打破数据孤岛，实现可控可监管的数据集共享机制。
* 对数据集采用链上+链下的存储模式:用户上传数据集，系统将数据集的摘要、哈希、数据集链外存储地址等元数据信息存储在链上；并将数据集主体基于IPFS等分布式存储技术存储在链外，由链上哈希验证链外数据是否被篡改；采用这种链上+链下的存储模式有效减少区块链系统的存储压力；采用AES对称加密算法对数据集进行加密后存储，用户下载数据集需要获取正确的加解密秘钥，提高数据存储的安全性。
* 系统采用区块链智能合约技术，将用户对数据集的上传、下载、更新等访问行为自动记录上链。基于区块链防篡改、可溯源特性，相关用户可对访问记录进行查询和可信溯源，实现对数据集整个生命周期的追溯和监管。
* 区块链智能合约技术能够自动对数据集在系统中的访问操作进行记录，在系统中实现可控可监管的数据共享；数据集在系统外的流转则采用数字水印等技术实现链外追溯，原型系统提供数字水印Demo接口，能够模拟水印添加和水印验证的过程。
* 采用MVC架构，将原型系统封装成用户友好的可视化界面，用户可基于Web应用与系统交互。

### 2. 系统架构

#### 2.1 运行时架构

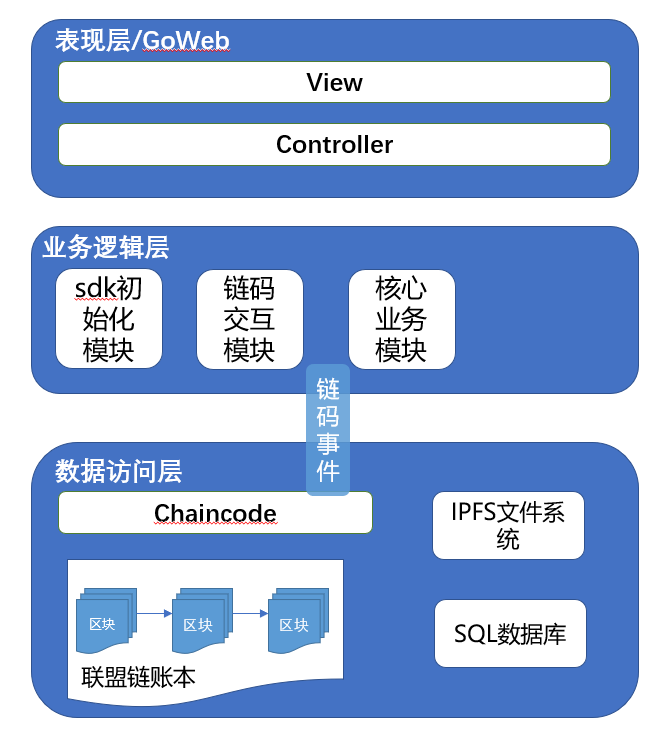
原型系统分为以下几个模块：Web客户端、Web服务器、IPFS文件系统、Fabric联盟链网络、Sql数据库。用户通过Web客户端向服务器端提交服务请求，服务端处理服务请求，并与Sql数据库、IPFS文件系统和Fabric网络交互，上传数据或获取响应数据；其中，用户数据（用户名、密码）和数据集加解密秘钥（由于区块链账本数据的公开性，不适合存储秘钥）存储在sql数据库中；数据集元信息（数据集名、摘要、IPFS地址、MD5哈希、时间戳等等）存储在联盟链账本上保证不可篡改；科学数据集文件存储在IPFS文件系统中，减轻区块链的存储压力，同时链下数据可由链上存储的哈希保证IPFS文件系统数据集的完整性。



**图1 系统架构图**

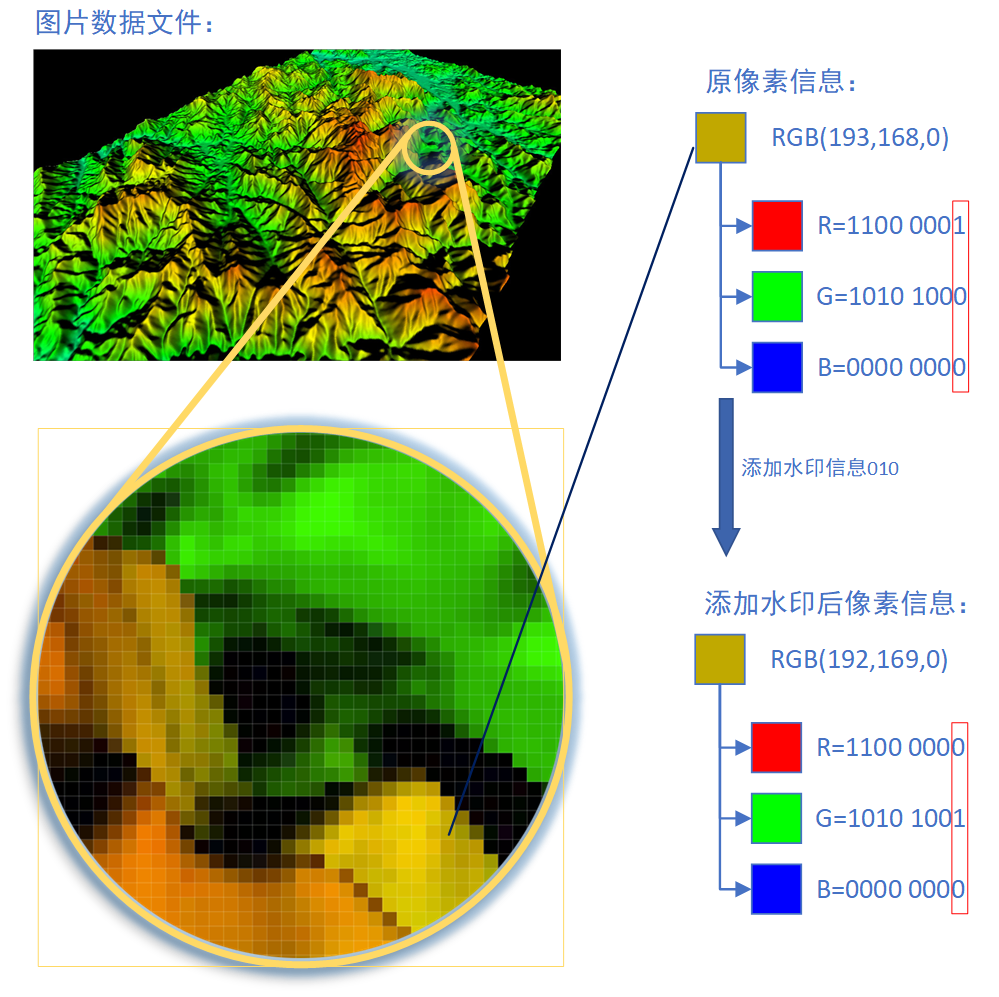
**2.2 业务逻辑架构**

数据共享溯源系统采用MVC架构设计，将系统分为表现层、业务逻辑层和数据访问层三层；表现层为用户提供Web UI，便于用户与系统的交互；业务逻辑层为系统主体实现，为表现层提供数据集上传、下载、查询、溯源等功能接口，并实现与区块链账本的交互；数据访问层使用区块链实现元数据的分布式可靠存储，并基于智能合约技术自动记录用户的访问记录，使用IPFS实现数据集主体的可靠存储，实现链上+链下的存储方式，有效减轻区块链的存储压力，使用SQL数据库存储注册用户信息以及数据集的加密秘钥信息，为用户托管数据加解密秘钥的分配和保存。系统的业务逻辑架构如图2所示。

**图2 业务逻辑架构图**

### 3. 详细设计

* 用户注册/登录功能：为实现本系统数据集的增删改查和溯源功能，系统需首先实现用户注册和登录的基本功能。注册用户名密码不允许为空字符串且用户名不能重复；用户名密码正确时登陆成功，并在服务器端设置session。
* 数据集注册：接收用户上传数据集信息，对参数名进行判断（数据集名和摘要不能为空，数据集名不允许重复）；系统为数据集分配AES加密秘钥对数据集进行加密，存储至IPFS文件系统，并基于FabricSDK交互，调用链码在联盟链账本上存储数据集的元信息；此外，由于加入区块链的节点均可获取区块链上的数据，基于区块链的这种数据公开性，显然将数据集加密秘钥存储在区块链状态账本上是不合适的，因此我们将数据集的加密秘钥存储至SQL数据库中，由服务方作为可信代理管理数据集加解密的对称秘钥。
* 数据集删除：本系统中数据集的删除本质上是设置删除标志位，使数据集禁止被下载。用户可改变自己数据集的下载状态（允许下载和禁止下载），该请求将调用链码修改该数据集在联盟链账本上的元信息（删除标志位），用户请求下载数据集时系统将从链上获取删除标志位的信息，决定是否同意用户的下载请求。
* 数据集更新：用户可以更新数据集的摘要信息和数据集文件信息，并对新的文件重新加密存储，并调用链码更新联盟链状态账本中的数据集元信息。
* 数据集查询：服务端调用智能合约，查询数据集元信息并返回给客户端。
* 数据集下载：用户可与数据集拥有者交易获取某数据集的秘钥，通过提交正确的秘钥下载数据。系统对秘钥进行验证，调用智能合约获取数据集的元信息（包含文件在IPFS存储的哈希地址），从IPFS文件系统获取数据集文件内容并用秘钥解密，最终将解密后的文件返回给客户端。
* 日志溯源：数据集的注册、更新、删除等数据集访问操作会在调用智能合约时被合约记录在联盟链账本上，用户可向服务器发送请求查询访问日志，服务端调用智能合约获取相关访问日志，并返回给用户。
* 数字水印：系统提供数字水印添加和检测Demo的接口，该数字水印Demo可以实现在分辨率为w\*h的非矢量图像中存储长度为w\*h\*3/8个字节的字符串且无法被肉眼所察觉。该方法的实现基于LSB（最低有效位，Least Significant Bit），具体原理如下：在非矢量图中，每个像素点都由数个通道中的数值来表示。如在常见的32位PNG格式图片中有R、G、B、A四个通道，分别表示红、绿、蓝、透明度，每个通道由8bit来表示。由于各个通道中的色彩是连续变化的，因此当表示像素色彩的数中最低位发生变化时，色彩的变化很小，人眼难以分辨的。因此我们可以通过这个原理在图片像素每个通道的最低位中隐藏1个bit的数据，将水印字符串嵌入图像中，且肉眼无法识别，数字水印添加示意图如图3所示。

**图3 数字水印添加示意图**

### 4. 重要数据表设计

* **SQL数据库表**：SQL数据库中保存两种类型的数据，包括用户数据（User表），用于用户身份认证；数据集加密信息（DataSetInfo），用于系统管理用户数据集的对称秘钥信息。XX

|  |  |  |
| --- | --- | --- |
| 字段名 | 说明 | 类型 |
| Id | 用户Id、主键、自增 | Uint |
| UserName | 用户名、非空、不允许重复 | String |
| PassWord | 密码、非空 | String |

**表1. User表**

|  |  |  |
| --- | --- | --- |
| 字段名 | 说明 | 类型 |
| Id | 数据集Id、主键、自增 | Uint |
| DataSetName | 数据集名、非空、不允许重复 | String |
| DataFileName | 数据文件名、非空 | String |
| EncryKey | 该数据集加解密的AES对称秘钥 | String |
| UserId | 数据集所属用户的Id、外键 | Uint |

**表2. DataSetInfo表**

* 联盟链状态账本数据表：重要的数据结构包括数据集元信息（MetaInfo表），用于在链上存储数据集的元信息；和操作日志（LogInfo表），用户记录数据集的操作日志，实现溯源和监管。

|  |  |  |
| --- | --- | --- |
| **字段名** | **说明** | **类型** |
| DataId | 数据集Id、主键 | Uint |
| DataName | 数据集名 | String |
| Abstract | 摘要 | String |
| Owner | 数据集拥有者 | String |
| Hash | 数据集哈希 | String |
| DataAddr | 数据集的IPFS哈希地址 | String |
| Timestamp | 时间戳 | String |
| DelFlag | 删除标志位 | String |
| TxId | 注册、更新数据集的区块链交易Id | String |

**表3.MetaInfo表**

|  |  |  |
| --- | --- | --- |
| **字段名** | **说明** | **类型** |
| DataId | 数据集Id、主键 | Uint |
| DataName | 数据集名 | String |
| Owner | 拥有者 | String |
| Operator | 操作者 | String |
| Operation | 操作（增、删、更新、下载） | String |
| TxId | 操作的交易Id | String |
| TimeStamp | 时间戳、操作时间 | String |

**表4. LogInfo表**