“统一区块平台”项目半年度总结

项目背景

4 月中旬与移动及第三方厂商一起参与了“茶叶溯源”项目的技术交流会议（见附录 1），会后移动大数据部门希望能将区块链技术在移动内部落地成应用或专利，作为今年的部分成果产出。在针对市面上多种区块链技术做了预研与对比（见附录 2）之后，我们的建议在移动内部先研发基于以太坊的“统一区块平台”而非“某具体区块链应用”，旨在将该技术落实为平台基础实施能力与对外提供上层应用服务的能力，类比云爬虫平台，简化区块链技术在实际场景中的应用，为其他传统项目/产品进行“区块赋能”。同时，在此平台之上开发一种基于区块链的数字资产交换装置，改变原有的生产关系，让数据所有者共同参与到整个数据价值变现的利益分配中，作为该平台的支持案例以便推广。（见附录 7）

项目主要工作内容和成果

- 基础设施研发：统一区块平台（80%），其中包含
  a) 区块平台架构设计（见附录 3）（100%）
  b) 区块平台模型设计（见附录 4）（100%）
  c) 区块平台开发语言及技术框架选型（见附录 2、附录 5）（100%）
  d) 区块平台环境搭建（见附件 1）（100%）
  e) 区块平台单元测试（该产品按 TDD 方法论进行）（90%）
  f) 区块平台基础开发（见附录 6）（90%）
     1. 平台层（100%）
     2. 共识层（80%）
     3. 支持层（按需支持各种组件，目前支持 H2/Hbase）
     4. 场景层/接口层（60%）
     5. 应用层（20%）
- 上层应用研发：图片存证与分享场景（20%）
  a) “图片存证与分享场景”应用开发（20%）
  b) 《一种基于区块链的数字资产交换方法及装置》专利编写（见附录 7、附件 2）（20%）

项目价值和效果

- 于项目：浙江移动区块技术能力落地
- 于产品：形成基础平台，定义任意数据上链范式
- 于团队：具备区块链应用实施能力

后续项目的规划和考虑

- 完成基础设施研发：统一区块平台（100%）
- 完成上层应用研发：图片存证与分享场景（100%）
- 协助产出相应专利：基于区块链的数字资产交换方法及装置（100%）

项目存在的主要问题

附录：

1. 茶叶溯源项目简介

该项目通过物联网技术及区块链技术，实现了将茶叶工艺流程中“采摘、晾青、杀青、揉捻、闷堆、发酵、干燥、精制”等环节细节及参数上链，继而通过包装上的二维码对每包出售的茶叶进行溯源操作，以便客户了解每包茶叶在制茶过程中的每个环节下的具体细节（如产地、照射时常、温度控制等），另外将扫码溯源的行为也上链，从而同时具备责任定位以及防伪确认的能力。由于项目涉及商业隐私，仅知道该项目底层区块链技术采用的是基于 IBM Hyperledger Fabric 的超级账本（属于联盟链）实现，中间层智能合约链码采用 Golang 及 Nodejs 语言编写，上层 web 应用后台使用 nodejs 开发。

2. 底层区块技术对比与选型

首先需要了解 IBM Hyperledger Fabric 基础架构与应用，以及作为智能合约的两个“头号人物”，它与另外一家“以太坊”的区别。抛开维护机构性质的异同（前者多家商业机构维护、后者社区维护，两者均开源），两者的差异实质体现在联盟链与公/私有链概念之上。

【私有链、公有链、联盟链的对比】

相同点：都在做分布式记账工作（节点间同步最新打包的区块信息，提高区块高度，维护公共账本），遇到记账冲突都有各自解决办法（半数以上节点可靠前提），从而保证数据不可篡改、全程可追溯。

不同点：联盟链的加入与写权限是私有的，只有读权限是对外开放，故不需要挖矿或代币（也就是 POW 工作量证明机制）来证明某个节点记账的有效性，信用背书依旧来自部署联盟链的一方或者多方各自的品牌与影响力，弱化了区块链去中心化的概念（但某种意义上更合规与安全）。公有链的加入与读写权限都对外开放，共识来源于各节点无法通过算法简化的硬性枚举算力工作（记账工作依赖于每个节点自身最近计算的结果），故记账需要各节点挖矿，需要具有代币机制来激励矿工（更多节点参与），在挖矿过程中消耗代币。私有链是不对外开放的公有链，一般作为开发共有链前的测试用途，或也可以视为经过权限控制的联盟链。

具体差异可参考以下表单：

Bitcoin	Ethereum	Hyperledger Fabric

Cryptocurrency required	Bitcoin	ether, user-created cryptocurrencies	none
Network	Public	public or permissioned	permissioned
Transactions	anonymous	anonymous or private	public or confidential
Consensus	proof of work	proof of work	PBFT
Smart contracts (business logic)	None	yes (Solidity, Serpent, LLL)	yes (chaincode)
Language	C++	Golang, C++, Python	Golang, Java

鉴于更成熟的社区生态系统以及提供更直观的合约部署方式，本平台采用以太坊最为初始底层实现方式，抽象上层区块特性，后续规划支撑不同的底层区块链版本，以不同的底层实现对上层应用提供区块服务，即 Blockchain As A Service(BAAS)。

3. “统一区块平台”架构设计概要

作为一个既能部署上层应用又可提供基础服务的平台，这里自底向上设计了相应层次：

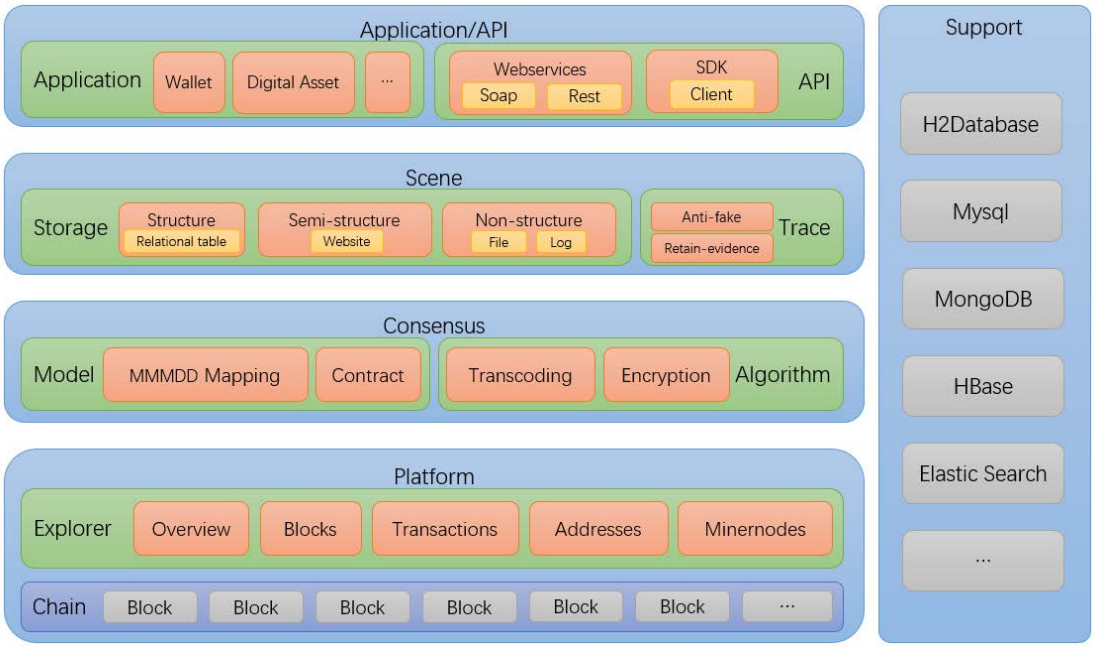
“平台层”(platform)：底层区块链平台，包含 WEB 界面与后端计划任务引擎，提供基础区块服务的能力，包含区块实时概览（区块高度、算力、难度、节点信息等）、分布式区块记账、矿工节点/同步节点管理与监控服务、区块监控与预览、交易监控与查询、关联地址明细查询、智能合约交互（部署、调用）、平台自检等能力。

“共识层”(consensus)：映射模型的智能合约层，提供了支持任意后端数据上链的“MMMDD”(metaKey->metaInfo->metaHash->dataValue->dataHash)共识映射模型与相关合约，以及基本加密算法。

“支持层”(support)：具体对接相应技术组件并各场景接口，如在存储业务中使用 H2Database/Mysql 实现结构化内容（关系表）上链、使用原生 FS/HBase/Mondb 实现半结构化/非结构化内容（日志、文件、音频、视频等）上链等。

“场景层”(scene)/“接口层”(interface)：面向各业务场景的抽象接口以及 soap/rest/sdk 方式的封装，如不同类型的存储（结构化/半结构化/非结构化）API。

“应用层”(application)/：“接口层”（API）：各具体业务的模块应用及对外的规范化接口，如：“资产存储与转移”应用/接口、“记账”应用/接口、“溯源”应用/接口、“公投”应用/接口、“考勤”应用/接口等。



4. MMMDD 模型设计概要(metaKey->metaInfo->metaHash->dataValue->dataHash)

设计这样一个模型，旨在解决三个痛点：

1. 保证上链数据的不可篡改性与可追溯性
2. 减少区块存储压力以及适应成员变量最大存储能力（以太坊只能合约变量最大只支持 32bytes）
3. 任意数据完整映射的上链能力

metaKey：目标数据指针的主键/序列号（上链，协同库记录）

metaInfo：目标数据指针的内容（如文件存储位置、文件存储时间、文件获取方式、文件加密方式等，根据实际数据类型选择不同后端，协同库记录）

metaHash：目标数据指针的摘要（上链，协同库记录）

dataValue：目标数据内容（如一段文本、一张图片，根据实际数据类型选择不同后端，相应协同后端存储）

dataHash：目标数据摘要（如文本/图片与主键/时间戳及相应参数 Base64 编码后的 MD5 值，上链，协同库记录）

注意：采用 UUID 作为 metaKey 而非 metaHash，是为了解决可能存在的摘要冲突问题（不同环境下的资源有着相同的指针信息或同一个环境下不同资源指针有着相同的摘要）。

5. 开发语言及技术框架选型

a) 开发语言选型

前端：html+css+js

后端：java+solidity+golang+shell

b) 技术框架选型

底层区块链实现：go-ethereum

智能合约链码开发：solidity

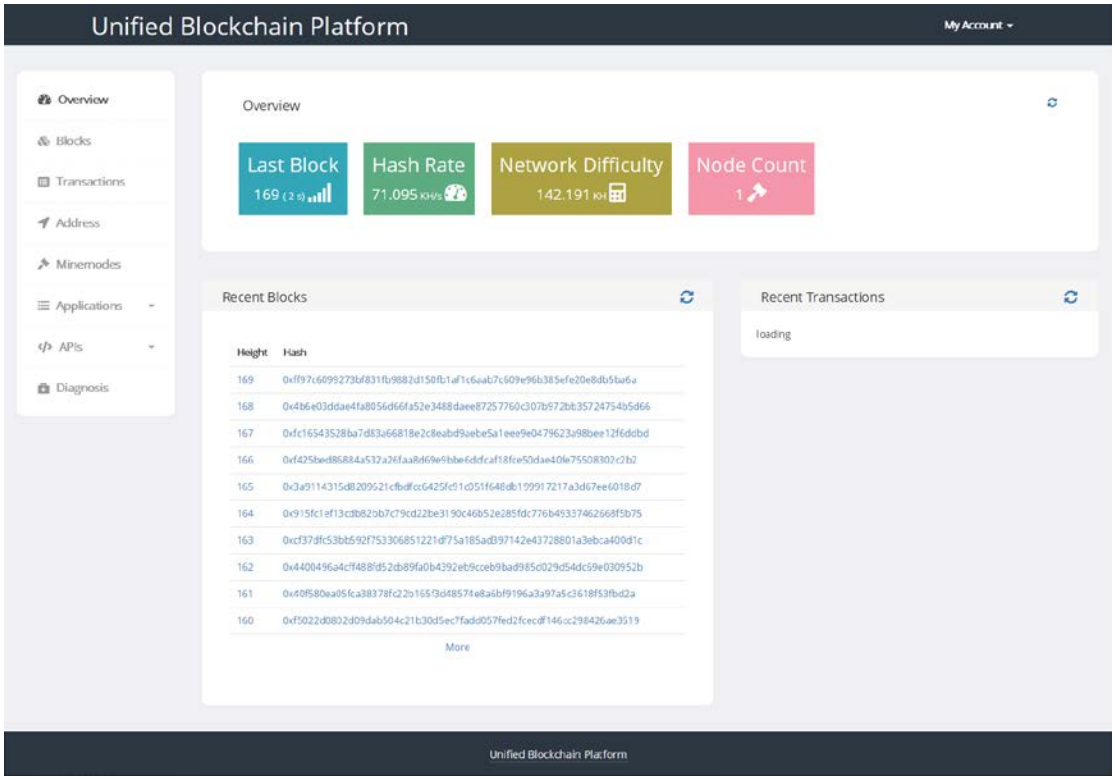
智能合约调度/部署及区块链交互：web3j

统一区块平台后端：spring+springmvc+shiro

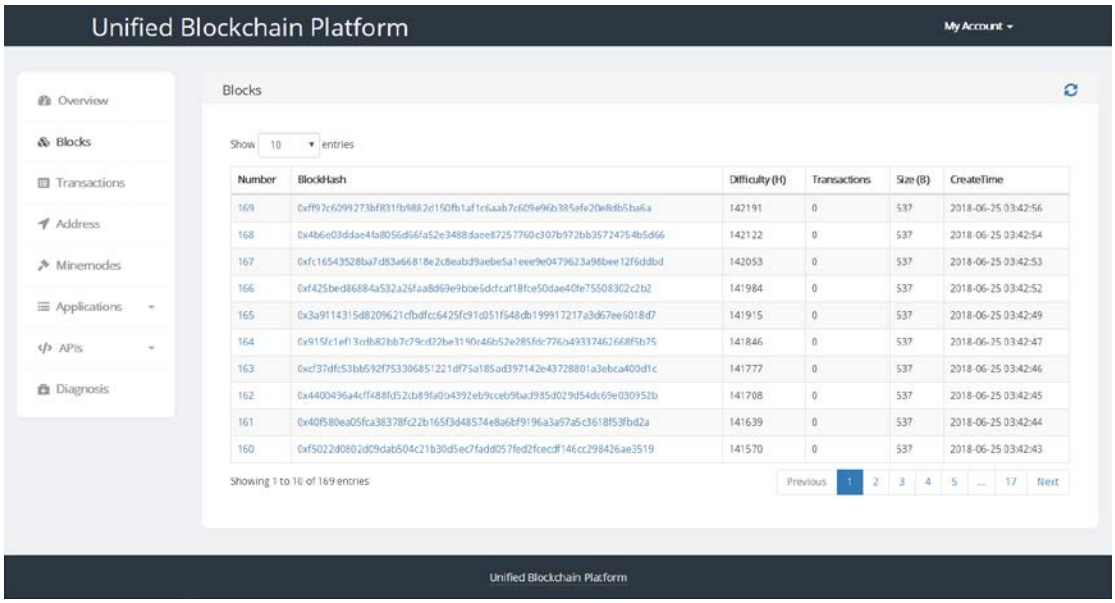
统一区块平台前端：jquery+bootstrap

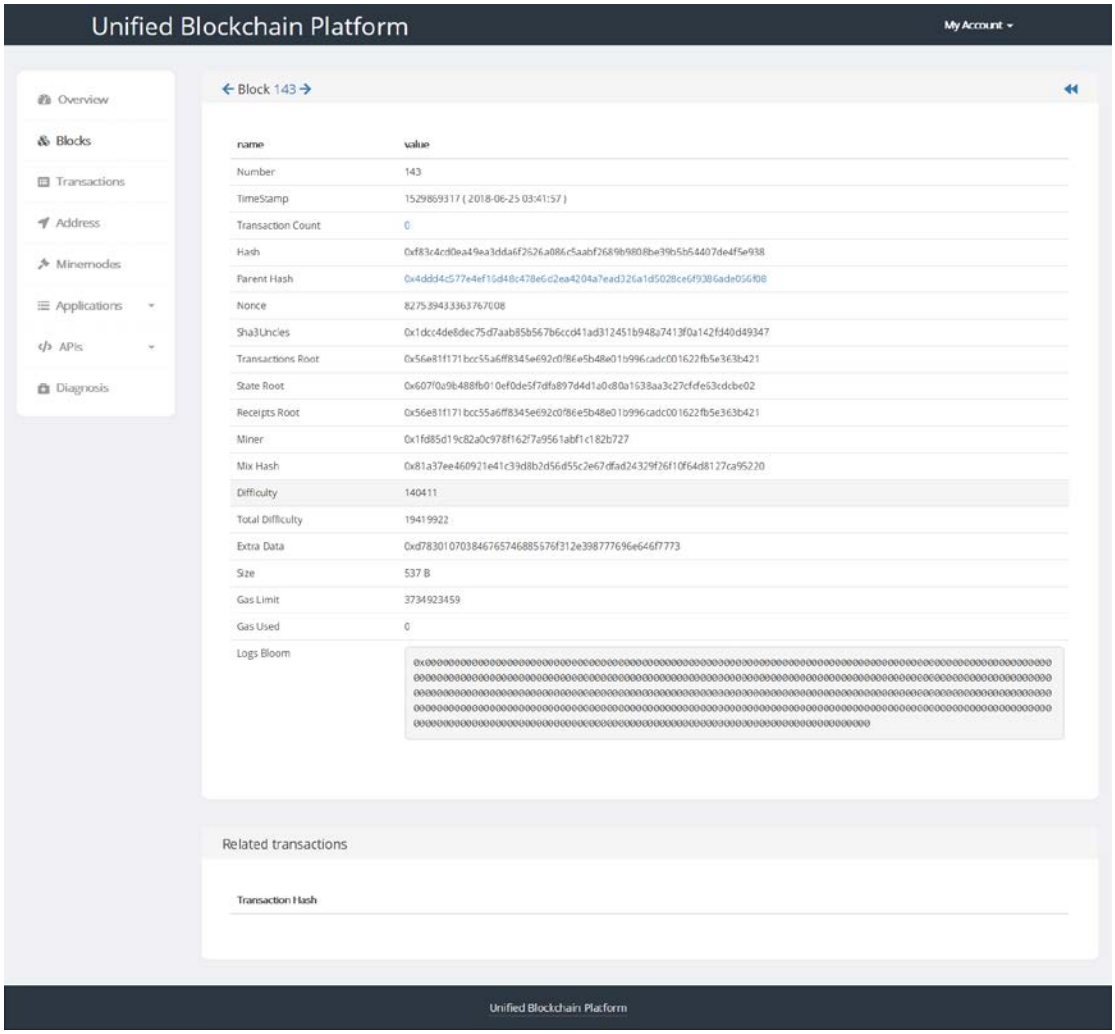
6. 平台部分功能展示

a) 区块实时概览（区块高度、算力、难度、节点信息等）

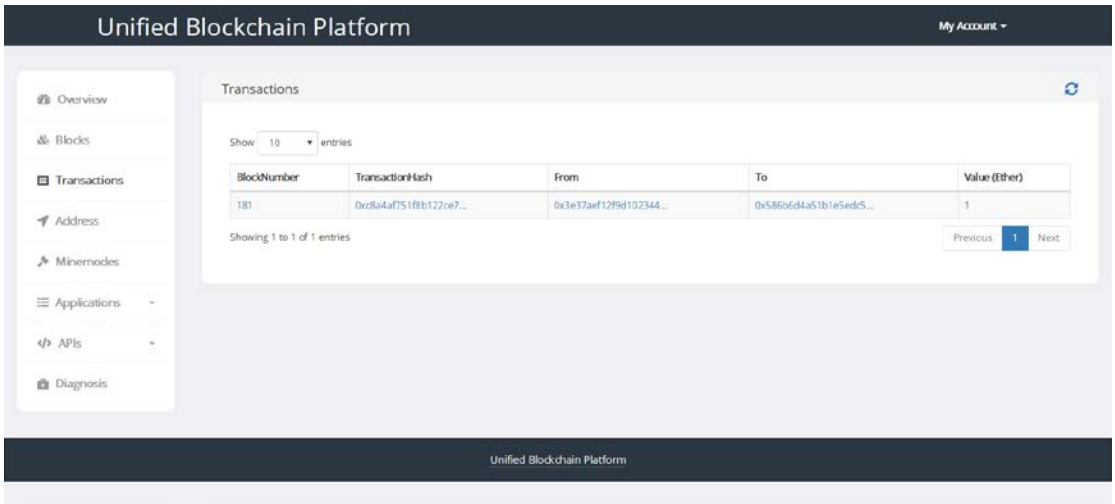


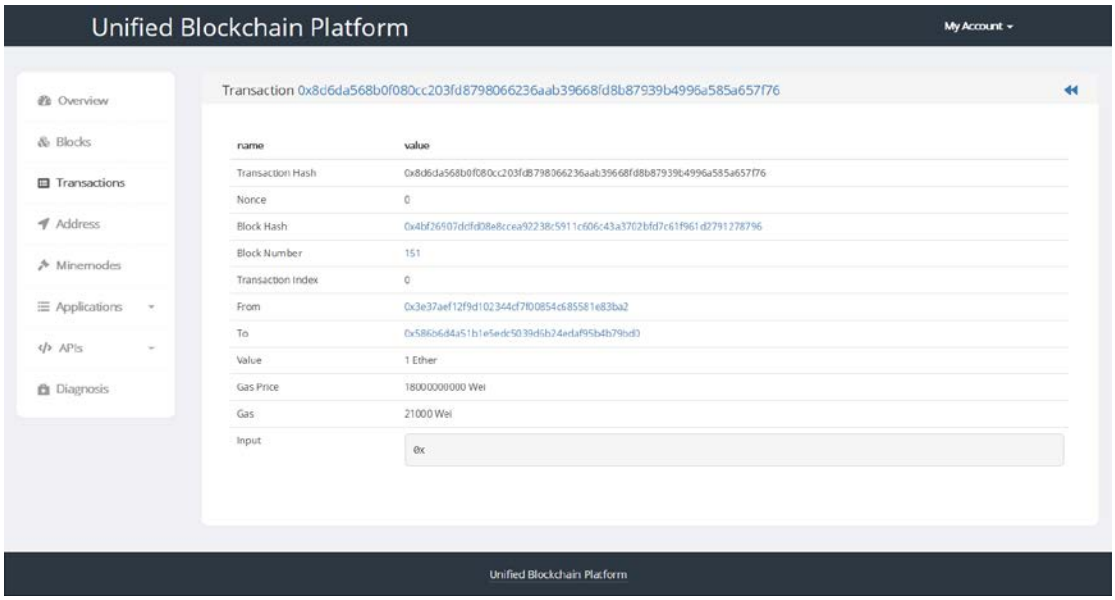
b) 区块监控与预览



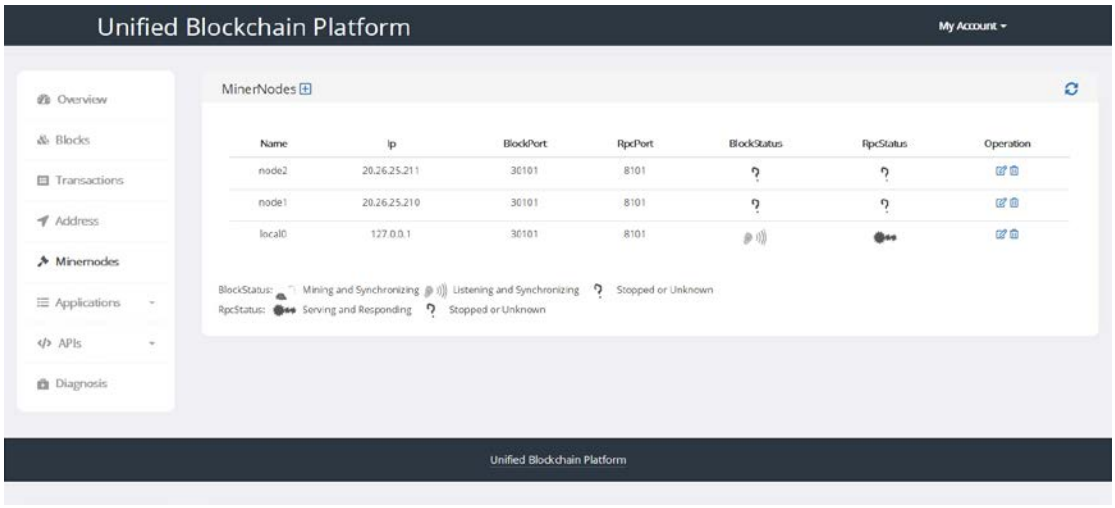

Unified Blockchain Platform
My Account
Overview
Blocks
Transactions
Address
Minernodes
Applications
APIs
Diagnosis
Block 143
name
value
Number
143
TimeStamp
1529869317 (2018-06-25 03:41:57)
Transaction Count
0
Hash
0xf83c4cd0ea49aa3dda6f2626a086c5aabf2689b9808be39b5b54407de4f5e938
Parent Hash
0x4ddd4c577e4ef15d48c478e6d2ea4204a7ead326a1d5028ce6f9386ade056f08
Nonce
8275394333637670O8
Sha3Uncles
0x1dcc4de8dec75d7aab85b567b6ccd41ad312451b948a7413f0a142fd40d49347
Transactions Root
0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421
State Root
0x607f0a9b488fb010ef0de5f7dfa897d4d1a0c80a1638aa3c27cfcfe63cdcbe02
Receipts Root
0x56e81f171bcc55a6ff8345e692c0f86e5b48e01b996cadc001622fb5e363b421
Miner
0x1fd85d19c82a0c978f162f7a9561abf1c182b727
Mix Hash
0x81a37ee460921e41c39d8b2d56d55c2e67dfad24329f26f10f64d8127ca95220
Difficulty
140411
Total Difficulty
19419922
Extra Data
0xd783010703846765746885576f312e398777696e646f7773
Size
537 B
Gas Limit
3734923459
Gas Used
0
Logs Bloom
Related transactions
Transaction Hash
Unified Blockchain Platform

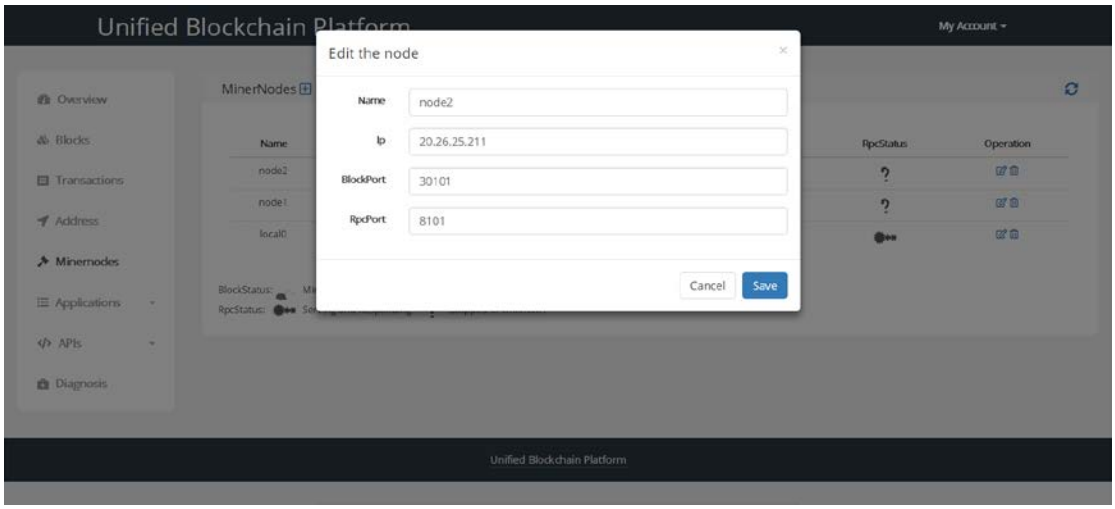
c) 交易监控与查询


Unified Blockchain Platform
My Account
Overview
Blocks
Transactions
Address
Minernodes
Applications
APIs
Diagnosis
Transactions
Show 10 entries
BlockNumber
TransactionHash
From
To
Value (Ether)
181
0xc8a4af751f8b122ce7...
0x3e37aef12f9d102344...
0x586b6d4a51b1e5edc5...
1
Showing 1 to 1 of 1 entries
Previous
1
Next
Unified Blockchain Platform

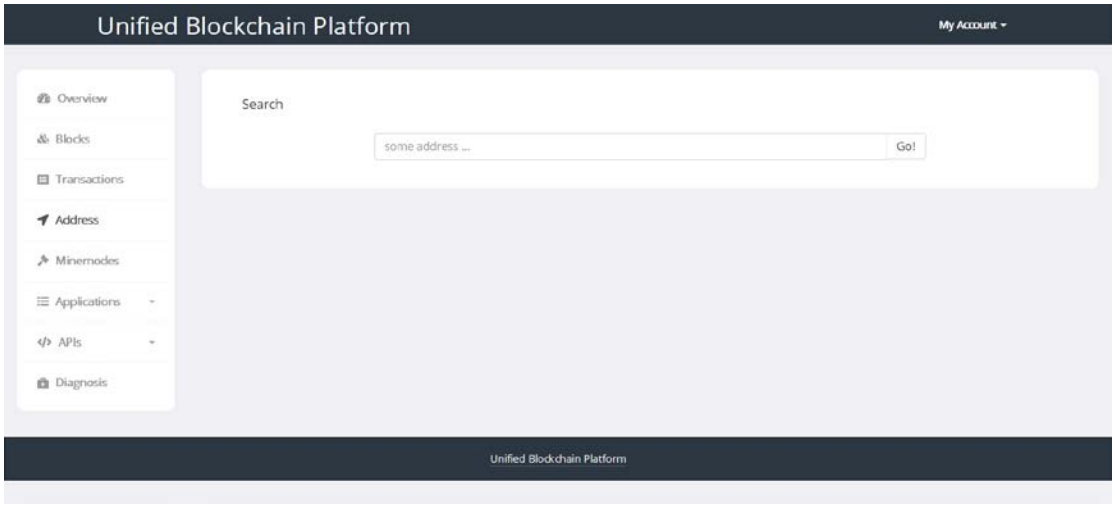


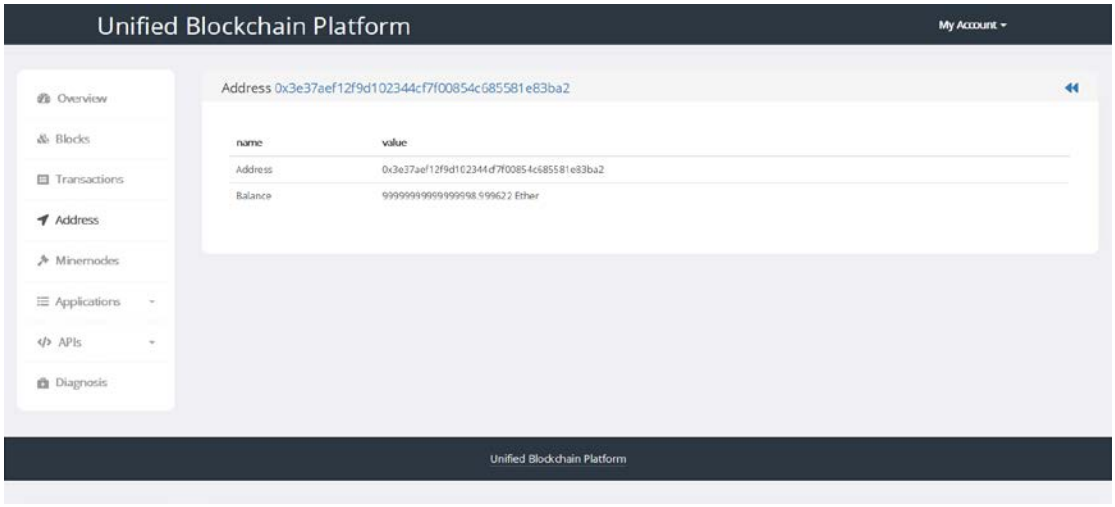
d) 矿工节点/同步节点管理与监控服务





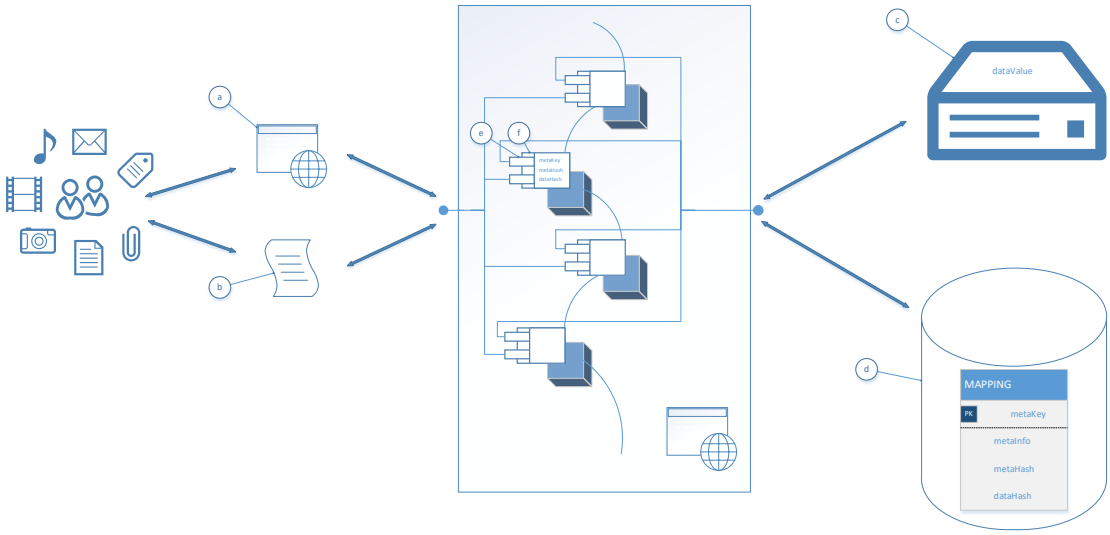
e) 关联地址明细查询部分





7. 一种基于区块链的数字资产交换方法及装置

本提案的主要目的通过增加基于区块链的部分装置改变原有的生产关系，让数据所有者共同参与到整个数据价值变现的利益分配中，改进后的数字资产交换流程图如下：



如上架构图中可以看到，新增了区块服务装置（包含底层区块服务、数据映射合约、区块浏览装置、数字资产交换应用等）用来支持完整的数字资产交换过程，其具体工作流程如下：

1） 用户主动上传(a)或在经过授权的情形下被动采集(b)数字资产（如文本文件、地理位置、操作行为等）；
2） 该数字资产以用户自身公钥加密，存储于服务端相关介质中(c)；
3） 该数字资产的存储描述信息（如存储方式、存储位置、加密标准、操作时间等）存储于协同库中(d)；
4） 该数字资产的存储描述的摘要记录于区块链中(e)；
5） 该数字资产加密后内容的摘要记录于区块链中(f)；
6） 资产原所属方使用私钥还原资产并使用目标方公钥加密资产并将新的描述摘要与加密后内容摘要记录于链上，从而实现资产转移。

（详见附件 2）