

金融服务平台——数据区块链存储

需求规格说明书

目录

- 一、概述(Summary)1
- 二、系统描述(System Description)2
 - 2.1 项目背景(Project Background)2
 - 2.2 项目功能(Project Function)2
 - 2.3 运行环境(Runtime Environment)2
 - 2.4 开发环境3
 - 2.5 条件与限制(Project Restriction)3
- 三、系统特性(System Feature).....5
 - 3.1 功能性需求(Functional Demand)5
 - 首先，本系统的区块统一存放在服务器端。5
- 四、外部接口需求(Outer Interface Demand)6
 - 4.1 硬件接口(Hardware Interface).....6
 - 4.2 软件接口(Software Interface)6
 - 4.3 通信接口(Message Interface).....7
- 五、其他非功能需求(Non-functional Demand)8
 - 5.1 性能需求(Performance Demand).....8
 - 5.2 安全设施需求(Security Facility Demand).....8
 - 5.3 安全性需求(Security Demand)8
 - 5.4 可拓展性需求(Expansion Demand)8
 - 5.5 合法性需求(License Demand).....8
 - 5.6 可靠性需求(Reliability Demand)8
 - 5.7 可测试性需求(Test Demand)8
 - 5.8 可维护性需求(Maintainence Demand).....9
- 附录 A：分析模型 (Appendix A: Model Analysis)10
 - 用例图 (Use case).....10
 - 数据流程图 (Sequence)11

一、概述(Summary)

1.1 项目目标(Project Goal)

此份软件需求分析报告是为**供应链平台-数据存储**编写的。它明确了所要开软件应具有的功能、性能，使系统分析人员及软件开发人员能清楚地了解需求，并且奠定了软件后续设计与开发工作的基础。

1.2 文档约定(Document Standard)

描述编写文档时所采用的标准为：

正文风格：宋体五号

提示方式：黑体加粗

重要符号：•

1.3 预期读者和阅读建议(Aiming Reader & Reading Suggestions)

本软件产品需求分析报告针对不同的预期读者，可能包括但不限于：**文档编写人员**，**开发人员**，**测试人员**。

1.4 参考资料(Reference)

软件需求分析报告：《软件工程实用教材》（吕云翔）。

核心技术说明：百度百科，Hyperledger Fabric 官方文档。

使用实例文档：《网上工作交易系统》需求分析规格说明书。

二、系统描述(System Description)

2.1 项目背景(Project Background)

区块链狭义上是一种按照时间顺序将数据区块以顺序相连的方式组合成的链式数据结构，并以密码学方式保证的不可篡改和不可伪造的分布式账本；广义上则是利用块链式数据结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、利用密码学的方式保证数据传输和访问的安全、利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算方式。区块链目前被广泛应用于金融领域，因为区块链最主要的特性是去中心化和不可篡改，而这两个特性完美契合金融领域的需求。

Hyperledger Fabric 是一种开源企业级许可分布式分类账本技术（distributed ledger technology, DLT）平台，专为在企业环境中使用而设计，与其他流行的分布式分类帐或区块链平台相比，可提供一些关键的差异化功能。其优点在于：1、是第一个支持在通用编程语言（如 Java, Go 和 Node.js）中创建的智能合约的分布式账本平台，而不是受限制的特定于域的语言（DSL）；2、获得了法律许可；3、支持可插拔的共识协议；4、不需要本机加密货币。

本项目将使用 Hyperledger Fabric 技术，专注于实现最终信息入链的功能，直接面向平台更上层的结构，并将为其提供对应接口。

2.2 项目功能(Project Function)

实名信息，合同签署信息，放款信息，还款信息的入链与查询。

2.3 运行环境(Runtime Environment)

硬件：

服务器：Xeon E5-2609 v3 1.9GHz 及以上，

内存：16GB DDR4 2400MHz 以上；

硬盘：至少 20G 以上；

软件：

操作系统：Windows 10

Docker: 17.06.2-ce 及以上版本

Docker Compose: 1.14.0 及以上版本

Java: Java Version 8

数据库: MySQL 8.0.16

2.4 开发环境

本项目采用 PC 端开发：

- 处理器型号：Intel 2.0GHz 及以上。
- 内存剩余空间：4G 及以上。
- 外存剩余空间：20G 及以上
- 网络配置：100M 网卡，串口。

软件：

- 操作系统：MacOSX, *nix, Windows 10
- Docker: 17.06.2-ce 及以上版本
- Docker Compose: 1.14.0 及以上版本
- Java: Java Version 8
- 数据库: MySQL 8.0.16

2.5 条件与限制(Project Restriction)

设计与实现：

- 在 Hyper Ledger Fabric 技术框架的基础上进行封装，区块链网络在 Docker 上运行，

chain code 选择 Java 作为开发语言；

- 使用 Rabbit MQ 作为消息队列，为上层提供接口；
- 要求遵循开发规范和虚拟账户系统的接口标准；
- 涉及到的所有数据统一使用加密的字符串进行传送；

假设和约束：

- 用户进行的操作包括融资、合同、放款、申请节点；
- 工期约束；

- 设备约束；

三、系统特性(System Feature)

3.1 功能性需求(Functional Demand)

首先，本系统的区块统一存放在服务器端。

本项目实现的功能为**实名信息**，**合同签署信息**，**放款信息**，**还款信息**的**入链与查询**。

在**授信企业**完成注册与**实名**后，平台完成审核并将该企业**实名信息入链**。

完成实名后，企业可以进行申请**融资**、**签署合同**、**充值**、**提现**等操作，后台可以进行**放还款**动作，根据企业的操作，**融资信息提交**、**签章成功返回**或者**账户余额变更**后将用户的**相关信息及变动记录**在**区块链**中，同时将服务器端的所有相关区块**同步信息**。

当企业或者政府等因故需要**查询**、**提取信息**的时候，可以在系统前端执行**查询**操作，账户及 workflow 管理系统**查询区块链中的信息**再返回相关信息。

账户及 workflow 管理系统应能**随时查询区块链中的信息**。

四、外部接口需求(Outer Interface Demand)

4.1 硬件接口(Hardware Interface)

暂无

4.2 软件接口(Software Interface)

• **Hyper Ledger Fabric:** 是一种开源企业级许可分布式分类账本技术（distributed ledger technology, DLT）平台，专为在企业环境中使用而设计，与其他流行的分布式分类帐或区块链平台相比，可提供一些关键的差异化功能。其优点在于：1、是第一个支持在通用编程语言（如 Java，Go 和 Node.js）中创建的智能合约的分布式账本平台，而不是受限制的特定于域的语言（DSL）；2、获得了法律许可；3、支持可插拔的共识协议；4、不需要本机加密货币。

• **Docker:** Docker 是一种执行操作系统级虚拟化的计算机程序，用于运行容器，让应用程序部署在容器下的工作可以自动化进行，借此在 Linux 操作系统上，提供一个额外的软件抽象层，以及操作系统层虚拟化的自动管理机制。

• **虚拟账户:** 在互联网平台注册时自行设置，是进行交易活动时的一个中介，需设置密码功能,通过查询虚拟账户余额，并用全额转回功能将虚拟账户内余额转回本人已在网上银行签约的同一身份证号下的活期账户。

本系统不与用户直接交互，所有接口设计皆基于与后台系统的交互。

用户以虚拟账户为载体，完成实名、申请融资、签署合同、充值和提现的操作。

用户在实名的时候，平台将用户信息入链，入链成功后本系统将返回一个标识用于标识用户。

用户申请融资以及签署合同的时候，平台先调用相关接口，提取用户的信息，将信息合并后再入链。

用户充值和提现的时候，平台先调用账户及 workflow 管理系统的相关接口，后台根据用户的标识从区块链中提取用户的信息，执行相应放还款操作后返回相应操作结果，若成功则变更账户余额，并将充值或提现信息入链。

所有入链后都将进行区块的同步操作。

序号	名称	备注
1	模块资源	
2	加密信息	

表 1 后台：信息存储接口

序号	名称	备注
1	节点标签	

表 2 后台：信息查询（提取）接口

4.3 通信接口(Message Interface)

消息队列（Message Queue）：是一种进程间通信或同一进程的不同线程间的通信方式，软件的队列用来处理一系列的输入，通常是来自用户。消息队列技术主要用于解决应用耦合、异步消息、流量削锋等问题。

五、其他非功能需求(Non-functional Demand)

5.1 性能需求(Performance Demand)

- 系统支持的并发操作数量：1000 次/s 以下；
- 响应时间：1s 级；
- 与实时系统的时间关系：异步；
- 容量需求：支持 1000GB 以下的数据量。

5.2 安全设施需求(Security Facility Demand)

- 区块链节点仅向系统后台授予数据访问权限。

5.3 安全性需求(Security Demand)

- 区块链链码完全封装，禁止外部与链码进行访问、交互、修改等行为。
- 由于系统业务的特殊性，需要有一定的抗攻击能力。
- 基于与上条同样的原因，各模块之间要相对独立，减少数据共享。

5.4 可拓展性需求(Expansion Demand)

- 本项目的代码实现应该力求模块化，使得功能拓展易于实现。

5.5 合法性需求(License Demand)

- 应保证本项目使用的所有软件、架构和技术均有合法授权或者为开源项目。

5.6 可靠性需求(Reliability Demand)

- 应保证储存模块在各类交易中均不出现系统崩溃、数据错误等异常情况。

5.7 可测试性需求(Test Demand)

- 任何模块的圈复杂度尽量不超过 15。
- 交付的系统必须通过 100%覆盖的单元测试。
- 开发活动必须使用回归测试，允许在 12 小时内重新进行完整测试。

5.8 可维护性需求(Maintenance Demand)

- 在收到一般的需求修改请求后，应该能够在 1-2 个工作日之内完成。对于重大需求或

者设计修改，应该能够在 1 周内完成。

- 对于 90%的代码 BUG，应该能够在 1 个工作日内予以修正。对于其他 10%的代码 BUG，应当能够在 3 个工作日内予以修正。

- 代码的圈复杂度应当在 10 以内。
- 任何对象的任何方法都尽量不超过 200 行代码。

附录 A：分析模型 (Appendix A: Model Analysis)

用例图 (Use case)

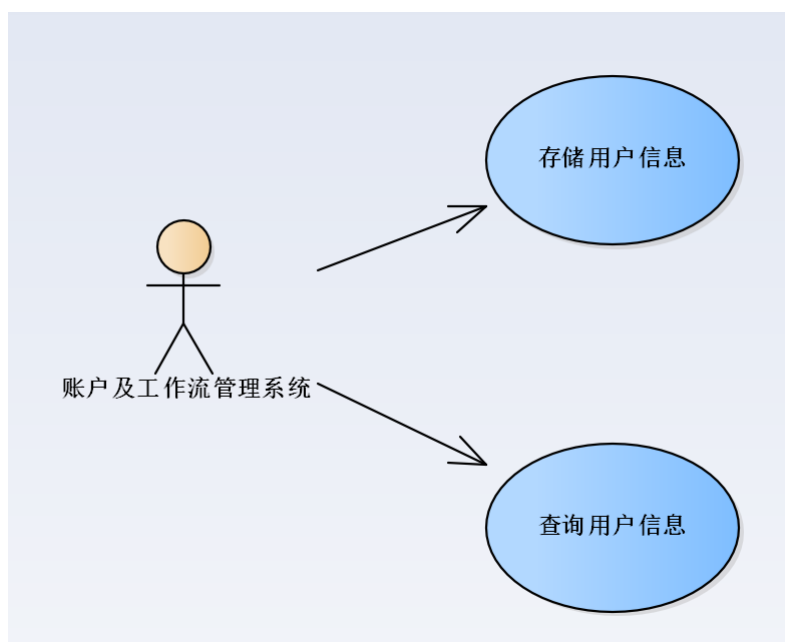


图 1 “金融服务平台-数据区块链存储”模块的用例图

用例账户及 workflow 管理系统，在接受到请求时，账户及 workflow 管理系统(虚拟账户)可以从本系统中查询用户信息，然后根据请求修改用户信息，再将修改后的用户信息写入区块链模块中。当一个企业完成了实名后，虚拟账户查询用户信息返回异常，因此创建一个新的节点用于存储该新用户。

数据流程图 (Sequence)

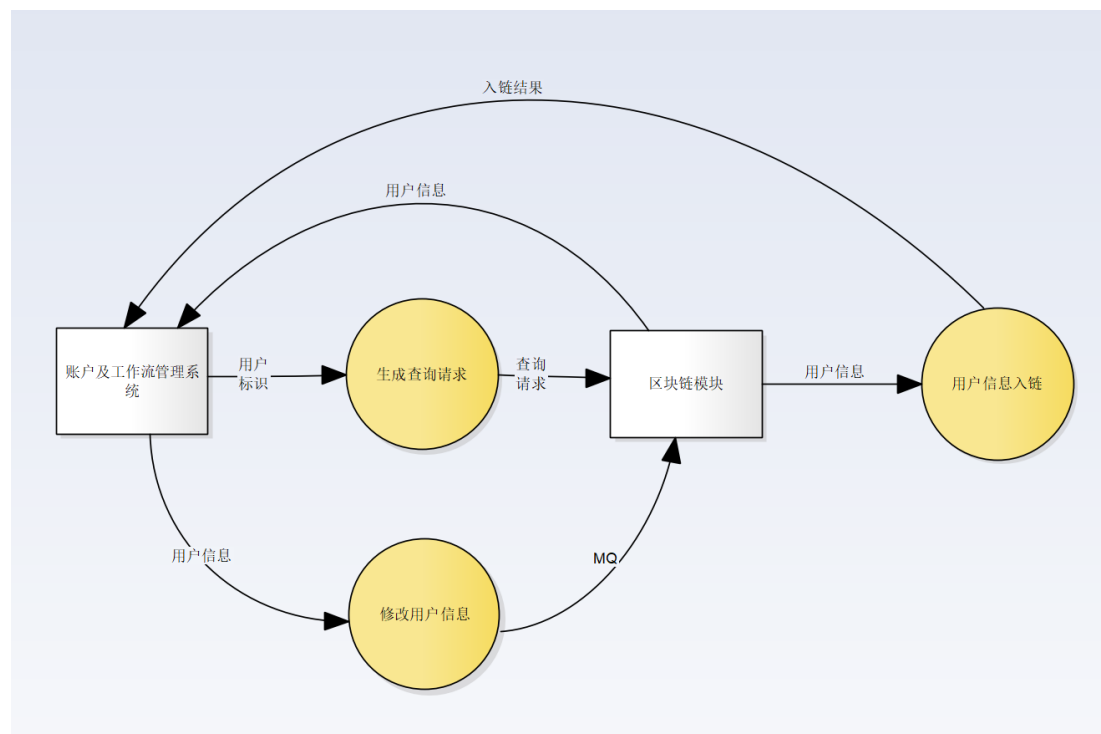


图 2 “金融服务平台-数据区块链存储”入链流程图

账户及工作流管理系统执行通过融资申请、完成签署合同、放还款的时候，利用用户标识生成查询请求发送给区块链模块，区块链模块获取消息后，返回区块中的用户信息，账户及工作流管理系统对得到的用户信息进行修改，将修改后的用户信息发送给区块链模块，区块链模块获取消息后，执行用户信息入链操作，并将入链结果返回给账户及工作流管理系统。