StarChain ICO系统

软件概要设计说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 文 档 编 号 |  | 保 密 等 级 | 机密 |
| 作 者 | 宛明 | 最后修改日期 | 2017.07.26 |
| 审 核 人 |  | 最后审批日期 |  |

**文档修订记录**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **章节名称** | **修订内容简述** | **修订日期** | **修订前**  **版本号** | **修订后**  **版本号** | **修订人** |
| 1 | 初稿 | 建立初稿 | 2017.07.26 |  | V0.1 | 宛明 |
| 2 |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |

备注说明：

1、提交全文时，请作者自行检查正文文字的字体、大小、颜色和格式。要求交付件文字为：宋体、五号、黑色、非斜体和非下划线。

目录

[1 简介 4](#_Toc488875225)

[1.1 目的 4](#_Toc488875226)

[1.2 范围 4](#_Toc488875227)

[1.3 术语定义 4](#_Toc488875228)

[2 总体设计 4](#_Toc488875229)

[2.1 逻辑架构图 4](#_Toc488875230)

[2.2 总体设计思路 5](#_Toc488875231)

[2.3 开发环境 7](#_Toc488875232)

[2.4 测试环境 7](#_Toc488875233)

[2.5 生产环境 8](#_Toc488875234)

[3 接口设计 9](#_Toc488875235)

[3.1 外部硬件接口 9](#_Toc488875236)

[3.2 外部软件接口 9](#_Toc488875237)

[4 数据库设计 9](#_Toc488875238)

[5 功能设计 9](#_Toc488875239)

[5.1 功能层次图 9](#_Toc488875240)

[5.2 功能清单 9](#_Toc488875241)

[5.3 协议类型管理 9](#_Toc488875242)

[6 用户界面规范设计 10](#_Toc488875243)

[6.1 协议类型管理 10](#_Toc488875244)

[7 安全性 10](#_Toc488875245)

[8 系统性能 11](#_Toc488875246)

[9 系统出错处理设计 11](#_Toc488875247)

[9.1 出错信息 11](#_Toc488875248)

[9.2 补救措施 12](#_Toc488875249)

# 简介

* 1. 目的

本文档旨在简要说明StarChainICO系统平台目前功能设计，作为开发配置规范的重要依据。

本文是在本系统需求详细分析、系统逻辑和物理架构的基础上，对整个系统软件实现进行详细分析后，按照系统要求的实现功能将系统划分为多个和功能相关的子系统，分别描述各个子系统的概要设计。本文同时定义了各个子系统之间的关系和详细接口，为下一步的编码实现提供设计框架。

本《概要设计说明书》读者对象为项目组全体开发人员和测试人员。

* 1. 范围

本文档覆盖了系统总体功能及功能说明，开发、测试环境。读者为业务制定人员、数据库设计人员、详细设计人员、开发人员、测试人员和项目相关手册编写人员。

* 1. 术语定义
* ICO：Initial Crypto Token Offering，初始加密代币发行
* 区块链：分布式账本系统
* 以太坊：在区块链上实现智能合约开源的底层系统
* 智能合约：
* SOA：Service Oriented Architecture，面向服务的架构模式
* 管理监控平台：WEB监控平台

# 总体设计

* 1. 逻辑架构图



* 1. 总体设计思路

系统从逻辑上可以分为以下几个部分：

应用模块：

存储模块：

网银对接：

区块链应用：

系统部署架构图如下所示：



应用服务器：部署包括AP1，AP2机器为应用服务器，采用负载均衡。

监控服务器：应用资源监控、报警。

数据库服务器：部署包括DB1和DB2机器为数据库服务器，采用主备。

云存储服务器：部署云存储。

运维监控云：云管理客户端。

服务请求方：均通过Nginx的负载均衡器访问应用。

* 1. 开发环境

软件环境:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **操作系统** | **版 本** | **说明** | **备注** |
| **1** | LINUX | Redhat V6.4 |  |  |
| **2** | 数据库 | Oracle11g |  |  |
| **3** | 开发工具 | Eclipse |  |  |
| **4** | JDK | JDK1.6 |  |  |

硬件环境:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **型号** | **数量** | **资源需求** | **操作系统** | **用途** | **备注** |
|  | 1 | 2核CPU以上，  8G内存以上，  300G存储以上 | Linux RedHat 6.4 | 数据库 |  |
|  | 1 | 4核CPU以上，  16G内存以上，  100G存储以上 | Linux RedHat 6.4 | 应用 |  |
|  | 1 | 2核CPU以上，  8G内存以上，  50G存储以上 | Linux RedHat 6.4 | 监控 |  |
|  | 1 | 4核CPU以上，  16G内存以上，  100G存储以上 | Linux RedHat 6.4 | 文件服务器 |  |

* 1. 测试环境

软件环境:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **操作系统** | **版 本** | **说明** | **备注** |
| **1** | LINUX | Redhat V6.4 |  |  |
| **2** | 数据库 | Oracle11g |  |  |
| **3** | 开发工具 | Eclipse |  |  |
| **4** | JDK | JDK1.6 |  |  |

硬件环境:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **型号** | **数量** | **资源需求** | **操作系统** | **用途** | **备注** |
|  | 1 | 2核CPU以上，  8G内存以上，  300G存储以上 | Linux RedHat 6.4 | 数据库 |  |
|  | 1 | 4核CPU以上，  16G内存以上，  100G存储以上 | Linux RedHat 6.4 | 应用 |  |
|  | 1 | 2核CPU以上，  8G内存以上，  50G存储以上 | Linux RedHat 6.4 | 监控 |  |
|  | 1 | 4核CPU以上,  16G内存以上，  100G存储以上 | Linux RedHat 6.4 | 文件服务器 |  |

* 1. 生产环境

软件环境:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **操作系统** | **版 本** | **说明** | **备注** |
| **1** | Linux | Redhat V6.4 |  |  |
| **2** | 数据库 | Oracle11g |  |  |
| **3** | 开发工具 | Eclipse |  |  |
| **4** | JDK | JDK1.6 |  |  |

硬件环境：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **型号** | **数量** | **资源需求** | **操作系统** | **用途** | **备注** |
|  | 2 | 4C16核  32G内存，  600G存储 | Linux RedHat 6.4 | 数据库 | RAC |
|  | 2 | 6C24核，  48G内存，  600G存储 | Linux RedHat 6.4 | 应用 | 通过F5  负载均衡 |
|  | 2 | 2核CPU，  8G内存，  200G存储 | Linux RedHat 6.4 | 监控 | 双机冷备 |
|  | 2 |  |  | 负载均衡 | 双机热备 |
|  | 2 | 4核CPU，  16G内存，  300G存储 | Linux RedHat 6.4 | 文件  服务器 | 通过F5  负载均衡 |

# 接口设计

* 1. 外部硬件接口

无

* 1. 外部软件接口

参见行内提供的接口文档。

# 数据库设计

具体参见《宁波通商银行ESB\_数据库设计说明书》。

# 功能设计

* 1. 功能清单
  2. ICO模块

### ICO预热

### ICO进行

### ICO项目管理

### ICO发起

功能描述：

用户自行发起项目ICO申请，每个用户不能重复发起，如果发起提示：“你有正在审核中的项目，请等待审核结果。如果有特殊需求，请联系我们客服团队。”

|  |  |
| --- | --- |
| **输入** | |
| 协议类型 | 协议类型名称 |
| 协议类别 | 协议类别（Server/Client） |
| 协议配置 | 协议配置(实现类、处理器、解析器) |
| 自定义协议配置 | 自定义协议配置（encoding、sign…） |
| **输出** | |
| ESB协议类型新建成功 | 保存 |

* 1. 交易平台

### 代币交易

功能描述：用户切换不同代币，进行对应的代币交易。

### 服务兑换交易

### 我的交易

功能描述：

* 1. 信息披露

功能描述：后台“信息披露管理”发布内容，该模块只负责展示。

* 1. 商城模块

（优先级三级，第三方产品先洽谈）

* 1. 我的信息模块

### 个人基础信息

功能描述：提供个人信息的更新，包括密码修改、手机号、邮箱。

### 交易明细

功能描述：展示交易流水信息

### 我的钱包

功能描述：提供人民币、eth、代币余额显示、充值、提现，一键兑换eth。

一键兑换需要展现当前汇率，使用定时爬虫任务，每1分钟抓取交易所汇率一次。前台页面以每半分钟请求后台获取当前汇率进行显示。

提供【全部兑换】功能。

人民币充值方案讨论：

1. 线上充值
2. 线下充值，后台自动处理（银行交易流水能否访问?）
3. 线下充值，人工处理：需要设置处理结果状态，并提供核账功能

### 我的项目

### 商品信息

功能描述：商品记录展示

### 代币地址管理

### 银行卡管理

功能描述：方案1：直连银行做验证和绑定，提供开户名、卡号、预留手机号，手机验证码是银行发送？

方案2：第三方接口，贵.

* 1. 社区模块

功能描述：帖子加精奖励代币

* 1. 系统管理模块

### 用户注册

功能描述：进行手机、邮箱注册验证、身份证信息录入，身份证图片上传，随机生成唯一的用户名(不可更改，用于系统管理，不需要展示)。

|  |  |
| --- | --- |
| **输入** | |
| **输入项** | **数据格式说明** |
|  |  |
|  |  |
|  |  |
| **输出** | |
|  |  |

### 用户登录

### 忘记密码

### 实名审核

功能描述：额外提供批量审核

### 用户权限管理

### 维护管理

### 日志管理

### 国际化设置

### 消息通知

### 内容管理

* 1. 客服系统

# 用户界面规范设计

* 1. 协议类型管理



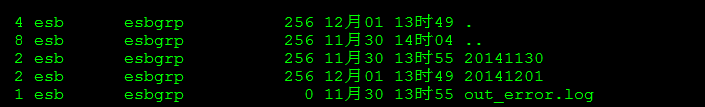
# 安全性

**系统安全：**

ESB系统将检查消费者系统服务调用端的IP地址是否有授权，存在授权服务请求正常处理；不存在授权或授权为拒绝，服务请求会被直接拒绝。所以，服务调用方系统要调用ESB，需要提交请求ESB系统的调用端IP地址，同时提供服务访问权限。

**日志安全：**

ESB系统代码使用行业标准API(log4j)工具包实现日志记录,对日志级别进行不同级别的设置与归档。



**PIN字段控制：**

银行关键性敏感字段需要加密控制，例如查询密码、交易密码、柜员密码或二磁道信息等，要求通过加密平台进行加密后设置在报文中才能进行系统间通讯。

服务调用方系统和服务提供方系统可以直接约定好加解密方式的情况下，ESB系统作为普通字段进行处理，不作特殊处理。

但是如果服务调用方系统和服务方系统不能直接约定好加解密方式的情况下，ESB系统需要进行PIN字段的转加密处理

# 系统性能

宁波通商银行对于ESB系统的性能要求如下：

|  |  |
| --- | --- |
| **序号** | **质量指标要求** |
| **1** | 提供日均2000万笔交易处理性能 |
| **2** | 单笔端到端服务请求消耗在企业服务总线上的时间要求小于30毫秒 |
| **3** | 保证系统在设计指标压力情况下的7×24小时长期稳定运行。 |

**提供日均2000万笔交易处理性能**

为满足ESB峰值日交易量2000万笔（依据系统交易运行规律80%的交易在20%的时间内完成的估算原则），所以系统的TPS至少达到1000才能够满足要求。

神州数码的ESB产品单台应用的处理性能就能够达到2000TPS以上，能够充分满足1000TPS的要求。

**单笔端到端服务请求消耗在企业服务总线上的时间要求小于30毫秒**

神州数码的ESB产品单台应用在100~300用户并发、TPS2600左右时，内部处理时间为20ms左右，充分满足宁波通商银行对响应时间的要求。

**保证系统在设计指标压力情况下的7×24小时长期稳定运行。**

神州数码的ESB产品自2008年11月份上线以来到目前为止，已经平稳运行7年，真正实现系统的永不停机服务，此过程还快速支持了银行新核心系统的投产和与国内某股份制银行的两行整合系统建设。

# 系统出错处理设计

* 1. 出错信息

错误信息对照表如下：

**desc="服务识别出错" esbcode="ESB-E-000045"**

**desc="前端拆包出错" esbcode="ESB-E-000048"**

**desc="后端拆包出错" esbcode="ESB-E-000048"**

**desc="前端组包出错" esbcode="ESB-E-000047"**

**desc="后端组包出错" esbcode="ESB-E-000047"**

**desc="前端数据映射出错" esbcode="ESB-E-000049"**

**desc="后端数据映射出错" esbcode="ESB-E-000049"**

**desc="系统识别出错" esbcode="ESB-E-000046"**

**desc="前端解密数据出错" esbcode="ESB-E-000033"**

**desc="前端加密数据出错" esbcode="ESB-E-000021"**

**desc="超出吞吐量" esbcode="ESB-E-000011"**

**desc="服务故障隔离" esbcode="ESB-E-000010"**

**desc="服务不在有效期" esbcode="ESB-E-000014"**

**desc="渠道故障隔离" esbcode="ESB-E-000035"**

**desc="渠道不在有效期" esbcode="ESB-E-000036"**

**desc="渠道已停止" esbcode="ESB-E-000025"**

**desc="前端读数据出错" esbcode="ESB-E-000042"**

**desc="后端写数据出错" esbcode="ESB-E-000043"**

**desc="后端读数据出错" esbcode="ESB-E-000042"**

**desc="核心路由调用超时" esbcode="ESB-E-000001"**

**desc="业务服务调用异常" esbcode="ESB-E-000002"**

**desc="业务服务已停止" esbcode="ESB-E-000006"**

desc="代理服务调用异常" esbcode="ESB-E-000004"

desc="身份验证不通过" esbcode="ESB-E-000016"

desc="核心路由未知异常" esbcode="OTHERS"

desc="前端未知异常" esbcode="OTHERS"

desc="后端未知异常" esbcode="OTHERS"

* 1. 补救措施

ESB平台通过负载均衡设备实现集群部署。每个ESB应用服务器构成一条完整的服务通路，服务通路之间功能独立，不会相互影响，可以有效的控制硬件故障的影响范围。

当某一路应用出现故障或者需要维护时，直接将这一路应用从集群中隔离，不影响整个集群的正常运行。

**探测过滤处理**：

http:在对外发布的url后面添加 **？DETECT=DETECT**

tcp:发送探测报文，**DETECT**

* 成功(健康)返回信息，默认返回SUCCESSED
* 失败(不健康)返回信息，默认返回FAILED