UtoDePIN开发文档

1.区块链网络

1.1L1节点

功能描述:处理区块链主网事务的节点。

1.2区块链网络

Fork以太坊:Fork以太坊（Ethereum），再根据项目需求进行优化和改造。

1.3创世区块

独立公链:创世节点独立分发。

开源:开源，2进制核对开源系统。

1.4网络切换

正式网络:正式网络。

测试网络:测试网络。

2进制核对开源系统:2进制核对开源系统。

1.5USDT跨链

第三方跨链系统接入公链底层:第三方跨链系统接入公链底层。

测试网络:测试网络。

2进制核对开源系统:2进制核对开源系统。

2.语言支持

繁体中文:繁体中文。

英文:英文。

2进制核对开源系统:2进制核对开源系统。

3.DAPP应用层

3.1模块

功能:模块功能描述。

原理:模块原理描述。

2进制核对开源系统:2进制核对开源系统。

3.2前端托管机制

允许DApp前端代码托管在区块链上:通过智能合约地址访问。

前端资源通过智能合约进行存储和分发:提供了一种去中心化的前端托管方案。

2进制核对开源系统:2进制核对开源系统。

3.3去中心化(DNS)DAAP

允许用户通过智能合约地址直接访问DApp:无需传统DNS。

DAAP协议通过区块链网络解析智能合约地址:实现去中心化的域名解析服务。

2进制核对开源系统:2进制核对开源系统。

3.4智能合约动态节点绑定

允许用户通过智能合约地址直接访问DApp:无需传统DNS。

智能合约根据后端节点的IP变动自动更新地址映射:保持用户访问的稳定性。

2进制核对开源系统:2进制核对开源系统。

3.5去中心化应用访问协议（DAAP）

允许用户通过智能合约地址直接访问DApp:无需传统DNS。

DAAP利用区块链网络提供去中心化的路由服务:增强了DApp的安全性和抗审查性。

2进制核对开源系统:2进制核对开源系统。

3.6区块链IP托管服务（BIHS）

将IP地址托管在区块链上:通过智能合约管理IP地址的分配和更新。

BIHS通过智能合约实现IP地址的动态管理:保障DApp后端的高可用性。

2进制核对开源系统:2进制核对开源系统。

3.7智能合约IP同步器

实时监测和同步智能合约关联的动态IP地址:同步器作为DApp的一部分，确保用户请求总是被导向最新的服务节点。

2进制核对开源系统:2进制核对开源系统。

3.8动态后端智能合约托管（DBSCH）

结合智能合约稳定性和动态IP灵活性:提供DApp托管解决方案。

DBSCH允许DApp在没有中心化后台服务的情况下运行:由智能合约自动管理后端功能。

2进制核对开源系统:2进制核对开源系统。

4.UTO区块浏览器

4.1主页

搜索框内容:输入地址/交易哈希值/区块哈希值/区块号/代币查询/钱包下ip/ip查询钱包/流量明细等数据。

首页顶部:UTO价格。

表格:UTO交易量（近7天）、流通总量、总市值、销毁总量、质押总量、验证人数。

最新区块:最新区块、区块号、出块时间、打包者、交易数。

查看更多:查看更多。

4.2最新交易

交易哈希值:最新交易。

从到:最新交易。

查看更多:查看更多。

4.3底部导航栏

开发者中心:开发者中心。

产品:UTO钱包、UTOwallet。

UTO:UTO、官网。

页面顶部的区块列表:区块列表（中文改为区块）、共xx条记录。

4.4区块列表

表格:时间、交易数、手续费。

打包者:奖励、时间、手续费。

4.5交易列表

页面顶部:交易列表（中文改为交易）、共xx条记录。

表格状态类型:已确认、打包中、成功、失败。

结果:发送人、接收人、金额、手续费。

4.6地址详情页

表格:时间、从到、状态、金额、手续费。

合约:合约概览、创建者、数量、其他资产、交易、基本信息、合约名称、合约代码、合约ABI、字节码。

4.7合约详情页

合约详情:基本信息、编译器版本、合约代码、合约ABI、字节码。

4.8交易详情页

概览:交易哈希值、发送人、时间、手续费、状态、接收人、转账金额、所在区块、GasPrice、GasLimit、Nonce、InputData。

4.9区块详情页

区块详情:区块哈希值、区块大小、区块高度、出块奖励、创建时间、出块者、交易、Gas消耗、GasLimit。

4.10代币详情页

代币详情页:代币详情页比合约详情页增加‘持有者’部分、前500名持有者、序号、地址、数量、占比、价值。

5.L2节点

5.1扩展节点

功能描述:处理快速结算与数据缓存。

6.模块功能描述

6.1发布任务

创建任务:创建任务。

铸造担保合约:铸造担保合约。

领取任务:领取任务。

锁定PCDN节点:任务开始。

任务结算:释放PCDN节点、执行担保合约、任务详情。

6.2后台系统

任务查询:任务查询。

6.3第三方PCDN

任务查询:模块、节点管理、添加节点、删除节点、节点详情。

6.4挖矿程序

识别用户ip:挖矿程序（算法整理钱包地址下的ip集）识别用户ip。

识别用户填入挖矿地址:ip流量显示。

该钱包地址收益明细展示:大数据整理求和数据。

6.5API对接数据

日总收益:AI系统接入计算。

IP流量明细:破解第三方广告程序集。

6.6IP对应钱包

大数据整理求和数据:链与第三方数据进行链侨对接。

6.7第三方破解

破解第三方广告程序集:自动签到、自动派发收益。

6.8第三方计费规则权重计算产量

破解第三方广告程序集:自动签到、自动派发收益。

6.9接口对接

链与第三方数据进行链侨对接:自动签到、自动派发收益。

业务池任务机制

1功能介绍

业务池是UtoDePIN网络网络的核心组件，提供集中的平台用于资源管理、任务分配和奖励分配。用户和企业可以在业务池发布任务，并设置完成任务的资源提供者的奖励。核心功能涵盖任务创建、资源匹配、奖励自动分配和任务执行监控。

2特点分析

业务池的特点包括去中心化结构、资源集中管理和奖励机制的有效激励。去中心化确保任务分配的公正性和透明度，集中管理提高资源利用率和执行效率，而奖励机制则鼓励资源提供者积极参与任务，加快任务完成。

3任务创建流程

任务创建始于用户或企业向业务池注入奖金需求，并定义任务目标和规格，包括任务内容、执行要求和完成时间。任务创建后，业务池负责发布任务，并提供详细信息给有兴趣的资源提供者。

4资源贡献者角色

资源贡献者在业务池中作为执行者，提供算力、存储和带宽等资源支持任务执行。通过API界面与业务池交互，贡献资源，参与任务执行，并根据贡献获得奖励。

5资金注入者职责

资金注入者通常是上游公司或投资者，负责为业务池提供资金支持，支付资源提供者的报酬和费用。根据任务性质和预期结果，向业务池注入资金，并在任务完成后根据表现支付报酬。

6任务执行与撤回

业务池监督任务执行过程，确保任务按既定目标和规格完成。若任务执行中出现问题或需求变更，提供任务撤回机制，允许任务创建者撤回和调整任务。

7应用领域

业务池应用领域广泛，包括PCDN加速、服务器加速与深度计算、人工智能等。在不同领域中，业务池通过提高资源管理效率，支持创新和发展。

双向挂单匹配系统（BidirectionalOrderMatchingSystem,BOMS）

运行原理

用户界面设计:系统提供界面，允许需求者和节点提供者输入报价和需求。

需求发布:需求者发布任务，明确要求、期望时间和支付价格。

节点报价:节点提供者根据资源情况提交价格报价。

智能匹配算法:系统根据需求者价格和节点提供者报价进行匹配。

双向报价机制:需求者和节点提供者可以双向报价。

实时更新与通知:系统实时更新挂单状态，并在匹配成功或新报价时通知用户。

交易确认与执行:匹配成功后，双方确认交易细节，智能合约自动执行任务分配和资金划拨。

任务进度跟踪:需求者跟踪任务执行进度。

评价与反馈机制:任务完成后双方互评。

智能合约保障:所有交易通过区块链智能合约执行，确保透明性、安全性和不可篡改性。

纠纷解决机制:提供纠纷解决机制，确保双方权益。

BOMS通过提供一个高效、透明的任务分配和资源匹配平台，促进需求者和节点提供者之间的交易，提高网络运作效率。

自定义95计费规则系统（Customized95BillingRuleSystem,C95BRS）

运行原理:

1. 任务池与资金池独立管理:

每个项目方拥有独立的任务池和任务资金池，确保资金的专款专用和任务的独立管理。

2. 时段定义与权重分配:

项目方可以根据自身业务特点定义不同的高峰时段，并为这些时段分配不同的权重，以反映不同时间段的重要性和成本。

3. 带宽使用量记录与处理:

系统会定期记录带宽使用量，并在高峰时段进行特别的处理，如降序排列和数据筛选，以确定计费带宽量。

4. 自定义计费公式编写:

项目方可以根据自己的需求编写计费公式，以适应不同的业务模式和成本结构。

5. 系数应用与日积分计算:

使用项目方定义的运营商系数和设备系数，结合设备在线时长，计算日积分。

6. 收益计算与分配:

根据日积分和高峰时段的权重，计算项目方在高峰时段的收益，并按照自定义规则进行分配。

7. 在线时长要求:

为了保证计费的准确性，设备需要保持全天在线。

8. 灵活的计费策略:

项目方可以根据自身业务的高峰时段和需求，灵活设置限速和结算比例，如30%~50%的收益分配。

9. 风险与收益评估:

项目方需要评估自定义计费规则可能带来的风险和收益，确保计费系统的可持续性和公平性。

10. 用户界面与反馈:

提供用户界面，使项目方能够轻松设置和管理自己的95计费规则，并接收用户反馈以优化计费策略。

通过C95BRS，UtoDePIN网络为不同的项目方提供了高度定制化的计费解决方案，以适应多样化的业务需求和市场条件。

7.区块链云盘板块

7.1功能描述

上传:用户可以将本地文件上传到云盘，支持断点续传和大文件上传。

下载:用户可以下载云盘中的文件，支持多线程下载以提高速度。

播放视频:云盘内的视频文件可以直接在线播放，无需下载。

外链分享:用户可以生成文件的外链，方便分享给其他人。

版本控制:云盘支持文件版本控制，可以查看和恢复旧版本的文件。

加密存储:所有文件在上传前都会进行加密处理，确保数据安全。

智能分类:云盘可以自动识别文件类型，并进行智能分类。

多设备同步:

智能搜索:提供强大的搜索功能，可以快速找到需要的文件。

多媒体预览:支持图片、音频和视频文件的预览功能。

回收站:删除的文件会被放入回收站，用户可以在一定时间内恢复。

8.Uto铸造合约

晚高峰日95计费系统

运行原理:

1. 时段定义:系统将每天的18:00至23:59定义为晚高峰时段，共计7小时。

2. 取样点记录:系统每5分钟记录一次带宽使用量，全天共记录288个点，晚高峰时段记录84个点。

3. 晚高峰时段处理:在晚高峰时段的84个带宽使用量点中，进行降序排列。

4. 数据筛选:从晚高峰时段的带宽使用量点中，去掉前5%（即4个点）。

5. 计费带宽量确定:在剩余的带宽使用量点中，选择最高的一个点的带宽使用量作为计费带宽量B。

6. 系数应用:使用运营商系数C和设备系数D，以及设备在线时长T，来计算日积分。

7. 日积分计算:根据日95计费公式\text(日积分)=B\timesC\timesD\times\left(\frac(T)(24)\right)进行计算。

8. 权重分配:晚高峰时段的权重设置为最高，通常超过90%，以确保收益最大化。

9. 在线时长要求:设备需要保持全天在线，以确保计费的准确性。

具体计算步骤:

1. 记录带宽使用量:系统每5分钟记录一次带宽使用量，全天共记录288次。

2. 晚高峰时段记录:在晚高峰时段（18:00至23:59）内，记录84次带宽使用量。

3. 排序带宽使用量:将晚高峰时段的84个带宽使用量点按照降序排列。

4. 去除前5%点:从排列好的带宽使用量点中去除前4个点。

5. 选择最大带宽使用量:选择去除后的带宽使用量点中的最大值作为B。

6. 计算日积分:使用公式\text(日积分)=B\timesC\timesD\times\left(\frac(T)(24)\right)计算日积分。

7. 计算收益:根据日积分和晚高峰权重，计算设备在晚高峰时段的收益。

8. 保持在线:确保设备在非晚高峰时段也保持在线，以满足计费需求。

显示节点当天现金收益

PCDN节点矿池机制：

1. 矿池聚合： 在PCDN节点矿池机制中，小节点用户将他们的资源集中到一个矿池中，共同参与挖矿活动。

2. 累计购买： 矿池会累积用户的贡献，直到总金额达到可以购买一枚代币的价值。

3. 奖励计算： 购买的代币数量除以用户贡献的价值，得出每个用户应得的代币数量。

4. 独立挖矿限制： 如果用户的贡献不足以达到一枚代币的价值增加10%的容差，他们将无法独立获得代币奖励。

5. 奖励发放： 只有当矿池累计的金额足够购买代币时，才会进行奖励的分配。

6. 优化等待： 用户可以通过提高自己的挖矿贡献来增加获得奖励的可能性，或者等待矿池累积更多的金额。

独立用户PCDN价值产出奖励

运行原理：

1. 价值计算： 用户PCDN产出的价值必须达到至少一枚代币的金额。

2. 容差升值： 系统为用户产出的价值增加10%的容差，用于升值，即允许价值在一定范围内波动而不影响奖励发放。

3. 手续费与返还： 类似于手续费，但这部分额外的价值最终会以某种形式返还给用户。

4. 奖励发放条件： 如果用户PCDN产出的价值未达到最低标准，系统不会立即发放奖励。

5. 等待与优化： 用户需要等待其PCDN产出的价值累积到满足条件，或者通过系统优化来提高产出价值。

分成条目：

产出价值标准： 用户PCDN产出的价值至少需达到一枚代币的金额。

容差升值： 系统自动为用户产出的价值增加10%的容差，用于升值。

手续费与奖励： 系统收取的类似手续费的价值最终会以某种形式返还给用户，增加用户的总收益。

奖励发放条件： 只有当用户PCDN产出的价值满足最低标准时，系统才会发放奖励。

等待机制： 如果产出价值未达标，用户需要等待价值累积或通过系统优化来提高产出。

优化建议： 用户可以通过优化PCDN配置或提高网络质量来增加产出价值，以满足奖励发放条件。

1. 初始条件：

总供应量：1亿枚

兑付储备金：1.01U

起始价格：1.01U

容差：1%

第1枚Uto代币购买：

任务池或矿池转入兑付储备金：1.01U + 0.01U（容差）= 1.02U

用户获得：1枚Uto

兑付储备金价值：1.01U + 1.02U = 2.03U

市场流通量：1枚Uto + 1枚Uto = 2枚Uto

计算当前价值 = 兑付储备金价值 / 市场流通量 = 2.03U / 2 = 1.015U

计算当前实际容差 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

计算总共实际容差 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

计算当前上涨率 = (1.015U - 1.01U) / 1.01U \* 100 = 0.495%

计算 总共上涨率 = (1.015U - 1.01U) / 1.01U \* 100 = 0.495%

计算当前购买成本 = 1.01U

计算累计购买总金额 = 1.02U

计算累计购买总数量 = 1

计算累计Uto总价值 = 1 \* 1.015U = 1.015U

计算立刻兑现亏损： 立刻兑现亏损 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

第2枚Uto代币

1. 初始条件更新：

总供应量：1亿枚

兑付储备金：2.03U（初始1.01U + 第一次购买的1.02U）

起始价格：1.01U

容差：1%

第2枚Uto代币购买：

任务池或矿池转入兑付储备金：1.015U + 0.01015U（容差）= 1.02515U

用户获得：1枚Uto

兑付储备金价值：2.03U + 1.02515U = 3.05515U

市场流通量：2枚Uto + 1枚Uto = 3枚Uto

计算当前价值 = 兑付储备金价值 / 市场流通量 = 3.05515U / 3 = 1.01838333U

计算当前实际容差 (1.02515 - 1.01838333) / 1.02515 \times 100= 0.66％

计算总共实际容差 =((2.04515 - 2.03676666) / 2.04515) \* 100 = 0.409913209％

计算当前上涨率 = (1.01838333U - 1.015U) / 1.015U \* 100 = 0.33%

计算总共上涨率 = (1.01838333U - 1.015U + 0.00495U) / 1.01U \* 100 = 0.83%

计算当前购买成本 = 1.015U

计算累计购买总金额 =1.02+1.02515=2.04515U

计算累计购买总数量 = 2

计算累计Uto总价值 = 2 \* 1.01838333U = 2.03676666U

计算立刻兑现亏损：立刻兑现亏损 = (1.02515U - 1.01838333U) / 1.015U \* 100 = 0.67%

第3枚Uto代币购买：

1. 初始条件更新：

总供应量：1亿枚

兑付储备金：3.05515U（初始1.01U + 第一次购买的1.02U + 第二次购买的1.02515U）

起始价格：1.01U

容差：1%

2. 第3枚Uto代币购买：

任务池或矿池转入兑付储备金：1.01838333U + 0.0101838333U（容差）= 1.0285671666U

用户获得：1枚Uto

兑付储备金价值：3.05515U + 1.0285671666U = 4.0837171666U

市场流通量：3枚Uto + 1枚Uto = 4枚Uto

3. 计算：

计算当前价值 = 兑付储备金价值 / 市场流通量 = 4.0837171666U / 4 = 1.0209292917U

计算当前实际容差 = (1.0285671666U - 1.0209292917U) / 1.0285671666U \* 100 = 0.67%

计算总共实际容差 = (4.0837171666U - 4.0695695833U) / 3.05515U \* 100 = 0.54%

计算当前上涨率 = (1.0209292917U - 1.01838333U) / 1.01838333U \* 100 = 0.63%

计算总共上涨率 = (1.0209292917U - 1.01U) / 1.01U \* 100 = 0.99%

计算当前购买成本 = 1.01838333U

计算累计购买总金额 = 2.04515U + 1.0285671666U = 3.0737171666U

计算累计购买总数量 = 3

计算累计Uto总价值 = 3 \* 1.0209292917U = 3.0627878750U

计算立刻兑现亏损：立刻兑现亏损 = (1.0285671666U - 1.0209292917U) / 1.01838333U \* 100 = 0.67%

去中心化交易所开发文档

1.引言

1.1目的

本文档提供了去中心化交易所(DEX)开发所需的详细信息，包括功能模块、用户界面设计、后端服务架构以及安全性设计。

1.2范围

本文档涵盖以下主要部分:

功能模块描述

用户界面(UI)设计

后端服务架构

安全性设计

2.功能模块

2.1首页

轮播图:用于广告宣传。

币种涨幅榜:展示币种波动情况，如AU、UTO、BTC、USDT、ETH。

快捷按钮:提供帮助中心、自选、充值、提现等快速入口。

公告栏:展示最新公告。

2.2领空投

项目方发布空投:项目方可以发布空投活动，并填写合约地址。

用户领取空投:用户可以领取空投，需要填写钱包地址并通过去中心化身份验证。

2.3行情

币种涨跌幅:实时获取币种价格和涨跌幅。

热门榜:向对付储备金转入一定金额可以上热门，金额越多排行越高。

审查榜:需要支付审查费用，通过DAO组织委员会审查是否是割韭菜体系。

2.4行情走势

曲线图（分时）:展示市场深度和实时成交情况。

K线图:提供市场深度和成交信息，以及币种的发行时间、发行总量、流通总量等简介。

2.5买入卖出

限时买入/卖出:用户可以设置期望的交易价格和数量，以及支付方式。

交易对:展示可交易的币种对。

挂单交易:用户可以查看已提交的买单或卖单列表。

3.用户界面设计

3.1首页设计

首页应包含所有快捷操作入口，以及实时更新的市场信息。

应有清晰的导航栏，方便用户快速找到所需功能。

3.2领空投界面

界面应提供简单的表单填写，以便用户领取空投。

需要有明确的提示信息，指导用户完成领取流程。

3.3行情界面

行情界面应提供实时的数据更新和图表展示。

应有筛选和排序功能，使用户能够根据需要查看不同币种的信息。

3.4交易界面

交易界面应简洁明了，方便用户快速下单。

应提供详细的交易对信息和交易历史记录。

3.5乌托邦Swap去中心化代币价值交换系统(基于指数上涨公式运转)

运行原理:

1.对付储备金与流通量:

系统维护一个对付储备金池和一个代币流通量。

代币的最低价值由对付储备金除以流通数量决定。

2.代币池子输入:

用户可以向代币池子输入代币数量，同时添加新发行的代币。

3.对付储备金池子输入:

用户可以向对付储备金池子输入代币数量，全网用户持有流通(建议项目方手上保留一枚) ，决定初始价值。(默认直接锁黑洞)

4.兑现价值:(添加池子的时候可以二选一)

模式一.用户持有的代币的兑现价值由对付储备金除以流通数量计算得出(低风险)。

买入代币的算法（起始流通量为1）

1. 初始化:

初始兑付储备金 R = 1.01U

初始市场流通量 C = 1

2. 计算每一枚代币的购买成本:

对于每一枚代币 i（从2到101，因为已经存在1枚）:

计算当前价格 P\_i = \frac{R}{C}

计算购买成本 \text{Cost}\_i = P\_i \times 1.01

更新兑付储备金 R = R + \text{Cost}\_i

更新市场流通量 C = C + 1

3. 计算总成本:

总成本 \text{Total Cost} = \sum\_{i=2}^{101} \text{Cost}\_i

4. 实际购买:

用户一次性支付总成本到兑付储备金 R\_{\text{final}} = R + \text{Total Cost}

更新市场流通量 C\_{\text{final}} = C + 100

5. 计算新的代币价格:

P\_{\text{final}} = \frac{R\_{\text{final}}}{C\_{\text{final}}}

卖出代币的算法

1. 初始化:

当前兑付储备金 R = R\_{\text{final}}

当前市场流通量 C = C\_{\text{final}}

用户要卖出的代币数量 m

2. 计算卖出金额:

\text{Amount} = P\_{\text{final}} \times m

3. 更新兑付储备金和市场流通量:

R' = R - \text{Amount}

C' = C - m

4. 计算新的代币价格（如果 C' > 0）:

P' = \frac{R'}{C'}

解释

R 和 C 分别代表兑付储备金和市场流通量。

P\_i 代表第 i 枚代币的当前价格。

\text{Cost}\_i 代表第 i 枚代币的购买成本，包括1%的容差。

\text{Total Cost} 代表购买100枚代币的总成本。

R\_{\text{final}} 和 C\_{\text{final}} 分别代表购买后的兑付储备金和市场流通量。

P\_{\text{final}} 代表购买100枚代币后的新代币价格。

m 代表用户要卖出的代币数量。

\text{Amount} 代表卖出 m 枚代币所得的金额。

R' 和 C' 分别代表卖出后的兑付储备金和市场流通量。

P' 代表卖出 m 枚代币后的新代币价格。

不支持传统LP流动性底池

模式二.灵活流动性交易系统 (Flexible Liquidity Trading System, FLTS)(风险中)

运行原理：

灵活流动性交易系统（FLTS）是一个创新的去中心化交易平台，支持用户根据个人偏好选择是否通过流动性池进行代币交易。以下是系统的主要运行原理：

1. 代币池子：用户可以向代币池子注入代币，为平台提供流动性，同时支持新代币的添加。

2. 对付储备金池子：用户可向此池子注入代币，以确定代币的初始价值。系统通过对付储备金与流通数量的比例来计算代币的初始价值。

最大卖出限制:因为除数和被除数不能为零，用户卖出最大到账99.(18)9%

3. 订单簿交易：用户可以通过订单簿挂单进行交易，系统支持买单和卖单，按照价格优先和时间优先的原则进行订单匹配。

4. 流动性池混合交易：用户可选择与流动性池进行混合交易，以快速匹配订单并成交。

5. 直接交易执行：用户可以选择直接交易模式，绕过流动性池，直接与订单簿上的对手方进行交易。

6. 容差调整：系统提供容差调整功能，用户可以设置容差大小，以调整交易的波动性和速度。

7. 智能合约驱动：所有交易都通过智能合约自动执行，确保交易的透明性、安全性和去中心化。

FLTS为用户提供了灵活性，允许他们根据市场条件和个人策略选择最佳的交易路径，无论是通过流动性池还是直接与订单簿上的对手方交易。

用户主动撤销流动性池再平衡机制（User-Initiated Liquidity Pool Rebalancing Mechanism）

运行原理：

1. 流动性池监控： 智能合约实时监控流动性池中的资产比例和总量，以确保交易的公平性和效率。

2. 用户撤销流动性： 用户可以发起撤销流动性的请求，希望提取其之前添加的流动性。

3. 条件触发平衡： 仅在用户撤销流动性时，智能合约才会触发平衡机制。

4. 资产比例计算： 智能合约根据用户最初添加流动性时的比例，计算应赎回的资产数量。

5. 多余资产分配：

如果因买入增多造成价格上涨，Uto份额会减少，USDT会增多。多余资产（如USDT）将按比例分配到其他流动性池中。

如果因卖入增多造成价格下跌，Uto份额会增多，USDT会减少。多余资产（如Uto）将按比例分配到其他流动性池中。

6. 部分赎回执行： 用户赎回其存入时对应的资产份额。

7. 缺少资产记录： 如果流动性不足，未能完全赎回的资产数量将被记录。

8. 后续交易补足：

当其他用户进行Uto卖出时，系统将缺少的Uto自动补足到之前添加流动性的用户钱包。

当其他用户进行Uto买入时，系统将缺少的USDT自动补足到之前添加流动性的用户钱包。

9. 智能合约自动执行： 所有操作均由智能合约自动执行，确保操作的透明性和自动化。

功能描述：

1. 流动性池监控： 智能合约实时监控流动性池中的资产比例，确保交易的公平性和效率。

2. 用户撤销流动性： 用户可以随时请求撤销其添加的流动性。

3. 条件触发平衡： 仅在用户撤销流动性时，触发智能合约的平衡机制。

4. 资产比例计算： 智能合约确保用户按照最初存入的比例赎回资产。

5. 多余资产分配： 多余资产被均分到其他流动性池中，以维持整个系统的流动性平衡。

6. 部分赎回执行： 用户获得与其存入比例相对应的资产。

7. 缺少资产记录： 未能完全赎回的资产数量被记录，等待后续交易补足。

8. 后续交易补足： 后续有用户进行相反交易时，系统将缺少的部分自动转到用户钱包地址。

9. 智能合约自动执行： 智能合约确保所有操作的透明性和自动化。

这种机制确保了用户在撤销流动性时能够获得其应得的资产份额，同时通过智能合约自动将多余资产分配到其他流动性池，并在后续交易中补足缺少的部分，从而保持了流动性池的整体平衡。

买入权重计算：

1. Uto池子分析：

系统首先分析Uto在流动性池和代币池中的分布情况，以确定哪个池子中的Uto份额较多。

2. USDT分配：

根据Uto的分布，系统将USDT按比例分配到对手储备金池和LP流动性池。大部分USDT将转入对手储备金池，以确保有足够的储备金支持Uto的买入。

3. LP流动性池补充：

剩余的少部分USDT将转入LP流动性池，以维持流动性池的深度和交易的即时性。

4. Uto转出：

同时，系统将按比例从代币池子和LP流动性池转出Uto到用户的钱包，完成买入交易。

卖出权重计算：

1. USDT池子分析：

系统分析USDT在流动性池中的份额，以确定卖出操作对流动性池的影响。

2. Uto转入：

根据USDT的份额，系统将按比例将Uto转入代币池子和LP流动性池，以补充因卖出操作而减少的Uto份额。

3. USDT转出：

同时，系统将按比例将USDT转出到用户钱包地址，完成卖出交易。

流动性池影响：

1. 流动性池深度：

LP流动性池的深度直接影响卖出操作的容差。流动性池越深，卖出操作的容差越低，即价格波动越小。

2. 兑付能力：

随着LP池子中资产的增加，系统的兑付能力增强，这降低了价格的跌幅度，因为市场有足够的流动性来吸收卖出压力。

3. 价格稳定性：

因此，LP池子添加的资产越多，系统对市场波动的缓冲能力越强，从而提高了价格的稳定性。

通过这些优化措施，FLTS系统能够更有效地管理流动性，确保交易的公平性和效率，同时为用户提供灵活的交易选择。智能合约的自动执行机制保证了所有操作的透明性和自动化，进一步增强了系统的可靠性和用户的信任度。

买入价格

[(对付储备金池÷全网用户持有流通数量包含LP池子里面代币)+LP流动性池价值÷2]+每枚代币设定的容差=买入的价格

卖出价格

[(对付储备金池÷全网用户持有流通数量包含LP池子里面代币)+LP流动性池价值÷2]-LP池子深度越高容差越低兑付能力越高=买出的价格]

模式三.采用传统Swap交互模型(亏损率超级高)

5.容差设置:(模式一，模式二都可以选择设定容差)(模式三属于自动容差)

用户可以设置每一枚币容差值，容差值越大，代币价值的涨幅越快。

静态容差容:差值的范围是从最低值精确到18位小数，到最高值50%。

动态容差

动态容差计算器

数学公式：

T(x) = \text{起始容差} + \left( \frac{\text{最终容差} - \text{起始容差}}{\text{最多代币容差} - 1} \right) \times (x - 1)

运行原理：

1. 定义参数：

\text{起始容差}：代币市场流通量的初始容差，例如10%。

\text{最终容差}：达到流动池流出90%时的容差，例如50%。

\text{最多代币容差}：达到最终容差时的代币数量，例如9000万枚。

2. 计算每枚代币容差增加量：

\text{每枚代币容差增加量} = \frac{\text{最终容差} - \text{起始容差}}{\text{最多代币容差} - 1}

3. 计算任意代币 x 的容差：

对于第一枚代币（x = 1），容差是 \text{起始容差}。

对于每增加的一枚代币，容差增加 \text{每枚代币容差增加量}。

4. 应用公式：

将 x（已发行的代币数量）代入公式，计算 T(x)。

示例应用：

起始容差：10%

最终容差：50%

最多代币容差：9000万枚

对于 x 枚代币：

T(x) = 10\% + \left( \frac{50\% - 10\%}{90,000,000 - 1} \right) \times (x - 1)

简化公式：

T(x) = 10\% + \frac{40\% \times (x - 1)}{89,999,999}

这个公式将根据已发行的市场流通量数量 x 动态计算容差 T(x)，从10%线性增加到50%，直到达到9000万枚代币。

注意：

如果 x 超过9000万枚，容差将保持在50%不变。

确保 x 不超过总发行量1亿枚。

模式一示例(未计算动态容差)

1. 初始条件：

总供应量：1亿枚

兑付储备金：1.01U

起始价格：1.01U

动态容差：1～20%(线性增加)

容差代币区间:2～9000万枚

转账机制:

1. 每次转账，10%的代币自动转入流动性池。

兑付机制:

1. 兑付储备金中的USDT将转入用户钱包。

交易示例:

1. 当前Uto的市场价值为2U/枚。

2. A用户向B用户转账100枚Uto。

3. 交易完成后:

B用户实际到账90枚Uto。

B用户获得20枚USDT

交易流程:

1. A用户发起100枚Uto的转账请求。

2. 系统自动计算10%的转账手续费，即10枚Uto转入流动性池。

3. 扣除手续费后，B用户实际收到90枚Uto。

4. 系统根据当前Uto的市场价值（2U/枚），计算出10枚Uto对应的USDT价值（20U），并转入B用户的钱包。

第1枚Uto代币购买：

用户转入兑付储备金：1.01U + 0.01U（容差）= 1.02U

用户获得：1枚Uto

兑付储备金价值：1.01U + 1.02U = 2.03U

市场流通量：1枚Uto + 1枚Uto = 2枚Uto

计算当前价值 = 兑付储备金价值 / 市场流通量 = 2.03U / 2 = 1.015U

计算当前实际容差 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

计算总共实际容差 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

计算当前上涨率 = (1.015U - 1.01U) / 1.01U \* 100 = 0.495%

计算 总共上涨率 = (1.015U - 1.01U) / 1.01U \* 100 = 0.495%

计算当前购买成本 = 1.01U

计算累计购买总金额 = 1.02U

计算累计购买总数量 = 1

计算累计Uto总价值 = 1 \* 1.015U = 1.015U

计算立刻兑现亏损： 立刻兑现亏损 = (1.02U - 1.015U) / 1.01U \* 100 = 0.495%

第2枚Uto代币

1. 初始条件更新：

总供应量：1亿枚

兑付储备金：2.03U（初始1.01U + 第一次购买的1.02U）

起始价格：1.01U

容差：1%

第2枚Uto代币购买：

用户转入兑付储备金：1.015U + 0.01015U（容差）= 1.02515U

用户获得：1枚Uto

兑付储备金价值：2.03U + 1.02515U = 3.05515U

市场流通量：2枚Uto + 1枚Uto = 3枚Uto

计算当前价值 = 兑付储备金价值 / 市场流通量 = 3.05515U / 3 = 1.01838333U

计算当前实际容差 (1.02515 - 1.01838333) / 1.02515 \times 100= 0.66％

计算总共实际容差 =((2.04515 - 2.03676666) / 2.04515) \* 100 = 0.409913209％

计算当前上涨率 = (1.01838333U - 1.015U) / 1.015U \* 100 = 0.33%

计算总共上涨率 = (1.01838333U - 1.015U + 0.00495U) / 1.01U \* 100 = 0.83%

计算当前购买成本 = 1.015U

计算累计购买总金额 =1.02+1.02515=2.04515U

计算累计购买总数量 = 2

计算累计Uto总价值 = 2 \* 1.01838333U = 2.03676666U

计算立刻兑现亏损：立刻兑现亏损 = (1.02515U - 1.01838333U) / 1.015U \* 100 = 0.67%

第3枚Uto代币购买：

1. 初始条件更新：

总供应量：1亿枚

兑付储备金：3.05515U（初始1.01U + 第一次购买的1.02U + 第二次购买的1.02515U）

起始价格：1.01U

容差：1%

2. 第3枚Uto代币购买：

用户转入兑付储备金：1.01838333U + 0.0101838333U（容差）= 1.0285671666U

用户获得：1枚Uto

兑付储备金价值：3.05515U + 1.0285671666U = 4.0837171666U

市场流通量：3枚Uto + 1枚Uto = 4枚Uto

3. 计算：

计算当前价值 = 兑付储备金价值 / 市场流通量 = 4.0837171666U / 4 = 1.0209292917U

计算当前实际容差 = (1.0285671666U - 1.0209292917U) / 1.0285671666U \* 100 = 0.67%

计算总共实际容差 = (4.0837171666U - 4.0695695833U) / 4.0837171666U \* 100 = 0.54%

计算当前上涨率 = (1.0209292917U - 1.01838333U) / 1.01838333U \* 100 = 0.65%

计算总共上涨率 = (1.0209292917U - 1.01838333U + 0.00495U + 0.00338U) / 1.01U \* 100 = 1.48%

计算当前购买成本 = 1.01838333U

计算累计购买总金额 = 2.04515U + 1.0285671666U = 3.0737171666U

计算累计购买总数量 = 3

计算累计Uto总价值 = 3 \* 1.0209292917U = 3.0627878751U

计算立刻兑现亏损：立刻兑现亏损 = (1.0285671666U - 1.0209292917U) / 1.01838333U \* 100 = 0.67%

以下是设立只涨不跌的测试代码，你可以运行一下(Python)

import time

from decimal import Decimal, getcontext

# 设置精确度为18位

getcontext().prec = 18

# 初始化参数

reserve\_fund = Decimal('1.01') # 兑付储备金

starting\_price = Decimal('1.01') # 起始价格

tolerance = Decimal('0.01') # 容差

total\_transactions = 1000000000 # 总交易次数，为了演示，这里设置为100次

# 初始化变量

current\_price = starting\_price

current\_reserve\_fund = reserve\_fund

current\_supply = Decimal('1') # 初始市场流通量已经有1枚Uto代币

user\_tokens = Decimal('1') # 用户手上持有的代币数量，初始为1

tokens\_to\_sell = Decimal('99.9999999999999999999999') # 每次卖出的代币数量

# 循环购买

for i in range(1, total\_transactions + 1):

# 计算买入前的价格

price\_before = current\_reserve\_fund / current\_supply

# 计算每次购买的金额（买入价格 + 容差）

buy\_amount = current\_price + (current\_price \* tolerance)

current\_reserve\_fund += buy\_amount # 更新储备金

# 更新市场流通量

current\_supply += Decimal('1')

user\_tokens += Decimal('1') # 用户每次购买1枚代币

# 计算买入后的价格

price\_after = current\_reserve\_fund / current\_supply

# 每满100次交易，卖出99.9999枚代币

if i % 100 == 0:

# 检查是否有足够的代币可以卖出

if user\_tokens >= tokens\_to\_sell:

# 卖出代币，减少储备金和市场流通量

sell\_amount = tokens\_to\_sell \* price\_after

current\_reserve\_fund -= sell\_amount

user\_tokens -= tokens\_to\_sell # 卖出99.9999枚代币

current\_supply -= tokens\_to\_sell # 减少市场流通量

else:

print(f"第{i+1}次交易：不足够代币卖出，当前持有量：{user\_tokens}, 需要卖出：{tokens\_to\_sell}")

user\_tokens = Decimal('0') # 如果没有足够的代币卖出，将用户持有量设置为0

# 计算总共上涨率

total\_rising\_rate = (price\_after - starting\_price) / starting\_price \* Decimal('100')

# 打印每次购买的结果

print(f"第{i+1}次购买：")

print(f"购买金额: {buy\_amount}")

print(f"买入前价格: {price\_before}")

print(f"买入后价格: {price\_after}")

print(f"总共上涨率: {total\_rising\_rate}")

print(f"用户持有量: {user\_tokens}")

print(f"兑付储备金金额: {current\_reserve\_fund}")

print("-" \* 50)

# 更新当前价格为买入后的价格，准备下一次交易

current\_price = price\_after

# 暂停0.1秒

time.sleep(0.1)

# 最终结果

print(f"最终兑付储备金价值：{current\_reserve\_fund}")

print(f"最终价值：{current\_price}")

print(f"最终总共上涨率：{total\_rising\_rate}")

print(f"最终用户持有量：{user\_tokens}")

print(f"最终累计Uto总价值：{user\_tokens \* current\_price}")

中文名:比特指数

英文简称:WBTCI

底池2100万枚

兑付储备金:0.00001634BTC

全网用户持有:0.00001634枚BTCI

价值计算:兑付储备金÷全网持有量=价值

容差算法:5%

Swap:模式二

买入1％滑点1％LP分红BTCI

卖出1%滑点1％LP分红BTCI

中文名:中文名称以太指数

英文简称:WETH

代币池300万枚

兑付储备金:0.000417747215256039ETH

全网用户持有:0.000417747215256039枚ETHI

价值计算:兑付储备金÷全网持有量=价值

容差算法:5%

Swap:模式二

买入0.5％滑点0.5％LP分红ETHI

卖出0.5%滑点0.5％LP分红ETHI

中文名称币安指数

英文简称:WBNBI

底池300万枚

兑付储备金:0.001758562248997812

全网用户持有:0.001758562248997812枚BNBI

价值计算:兑付储备金÷全网持有量=价值

容差算法:5%

Swap:模式二

买入0.5％滑点0.5％LP分红BNBI

卖出0.5%滑点0.5％LP分红BNBI

等

4.后端服务架构

4.1数据库设计

需要设计一个能够处理高并发请求的数据库。

数据库应支持快速查询和数据更新。

4.2API设计

API应提供完整的端点，以支持前端的所有功能。

应有安全措施，如认证和授权，保护API端点。

4.3服务器和网络

服务器应有足够的处理能力和存储空间。

网络应设计为高可用性和安全性。

5.安全性设计

5.1加密和认证

所有敏感数据应进行加密处理。

用户认证应采用多因素认证方式。

5.2防御机制

系统应具备防止DDoS攻击的能力。

应有定期的安全审计和漏洞扫描。

5.3合规性

平台应遵守相关的法律法规和行业标准。

应有透明的隐私政策和用户协议。

6.挂单交易订单簿

6.1概述

挂单交易订单簿是交易平台的核心功能之一，它允许用户根据市场条件设置买卖订单。用户可以指定交易的价格和数量，订单将按照用户设定的条件在订单簿中排队等待成交。

6.2功能描述

挂单交易:用户可以设置一个期望的交易价格，并在该价格达到时自动成交。

订单匹配:挂单交易不会直接进入流动池，而是在订单簿中等待匹配。

灵活性:用户可以更灵活地控制交易时机和价格。

6.3卖单挂单

设置卖单:用户可以设置一个卖单挂单，即在希望卖出的价格上挂出订单。

等待成交:订单将保留在订单簿中，直到市场价格达到用户设定的卖出价格。

市场条件:此功能允许用户在不立即交易的情况下，等待市场达到预期的价格。

6.4买单挂单

设置买单:类似地，用户也可以设置买单挂单，即在希望买入的价格上挂出订单。

市场下跌策略:这为用户提供了一种策略，以等待市场下跌至其期望的买入价格。

6.5订单管理

查看订单:用户可以在订单簿中查看所有当前挂单的状态，包括自己的挂单。

取消订单:用户可以随时取消未成交的挂单。

订单更新:用户可以修改已有挂单的价格和数量，以适应市场变化。

6.6技术实现

数据库设计:订单数据需要存储在高性能的数据库中，以支持快速的查询和更新。

前端展示:订单簿应在前端以列表或图表的形式展示，方便用户查看和操作。

后端逻辑:后端服务需要处理订单的创建、匹配、成交和取消等逻辑。

6.7安全性考虑

认证授权:确保只有订单的所有者可以查看、修改或取消订单。

数据加密:订单数据在传输和存储过程中应进行加密，以防止数据泄露。

6.8用户界面设计

直观展示:订单簿应直观展示所有挂单，包括价格、数量、订单状态等信息。

操作便捷:用户应能轻松地进行挂单、取消和修改操作。

钱包功能开发文档

1.引言

1.1目的

本文档旨在为去中心化交易所的钱包功能开发提供详细指南。钱包功能是用户管理其加密资产的核心工具，需要确保安全性、易用性和功能性。

1.2范围

本文档包括钱包功能的以下主要部分:

钱包创建与管理

导入与导出钱包

转账功能

资产管理

安全性设计

2.钱包创建与管理

2.1创建钱包

功能描述:用户可以创建新的钱包，生成钱包名称、密码和备份助记词。

用户界面:提供简洁明了的界面，引导用户完成创建过程。

安全性:助记词需安全生成和备份，密码需加密存储。

2.2钱包管理

功能描述:用户可以管理其钱包，包括查看钱包地址、余额和交易历史。

用户界面:提供清晰的钱包信息展示和操作选项。

安全性:确保所有敏感信息都经过加密处理。

3.导入与导出钱包

3.1导入钱包

功能描述:用户可以通过助记词或私钥导入已有钱包。

用户界面:提供导入选项，引导用户输入助记词或私钥。

安全性:确保导入过程中的敏感信息安全。

3.2导出钱包

功能描述:用户可以导出其钱包的助记词或私钥，以便在其他设备上使用。

用户界面:提供导出选项，允许用户选择导出格式。

安全性:导出过程中需提醒用户注意信息安全。

4.转账功能

4.1发起转账

功能描述:用户可以向其他地址转账，输入转账金额、收款地址和矿工费。

用户界面:提供直观的转账表单，包括地址验证和交易确认。

安全性:交易前需进行多重验证，确保交易安全。

4.2交易历史

功能描述:用户可以查看其钱包的交易历史记录。

用户界面:以列表形式展示交易历史，包括交易详情和状态。

安全性:交易历史信息需加密存储，确保用户隐私。

5.资产管理

5.1资产概览

功能描述:显示用户钱包中的所有资产及其价值。

用户界面:提供资产列表，包括资产名称、数量和价值。

安全性:资产信息需实时更新，确保数据准确性。

5.2资产操作

功能描述:用户可以进行资产的划转、冻结和解冻等操作。

用户界面:提供操作选项，引导用户完成相应操作。

安全性:所有资产操作需进行身份验证和权限检查。

6.安全性设计

6.1加密技术

功能描述:使用行业标准的加密技术保护用户数据。

技术实现:采用AES、RSA等加密算法对敏感数据进行加密。

6.2认证机制

功能描述:实施多因素认证，确保用户身份安全。

技术实现:结合密码、短信验证码、生物识别等多种认证方式。

6.3交易安全

功能描述:确保交易过程的安全，防止欺诈和盗窃。

技术实现:交易过程中进行多重验证，如交易密码、电子邮件确认等。

6.4安全审计

功能描述:定期进行安全审计，发现并修复潜在的安全漏洞。

技术实现:采用自动化工具和手动审计相结合的方式。