# 软件设计说明书

## 前言

PDX TeX (Token Exchange) 是构建于 PDX Utopia 区块链协议栈基础之上的安全的 ERC20规范的虚拟资产交换平台，以智能合约的形式对外提供提供服务接口，以完全去中化方式运行，安全稳定，可完全防止传统中心化交易所的各种数据作弊手段。

PDX TeX 被设计为可插拔模块并兼容 Ethereum 协议，兼容 Solidity ABI 协议，符合 Web3.js 接口规范，提供友好的 Dapp 开发者接口；

软件主要开发语言基于 golang 和 solidity ，对于终端接入方还会用到 java、nodejs 等完成对接。

## 目标

1、提供 ERC20 合约注册表，在合约部署时系统自动拦截 ERC20 合约并将地址信息填入注册表，并对外提供获取注册表信息功能；

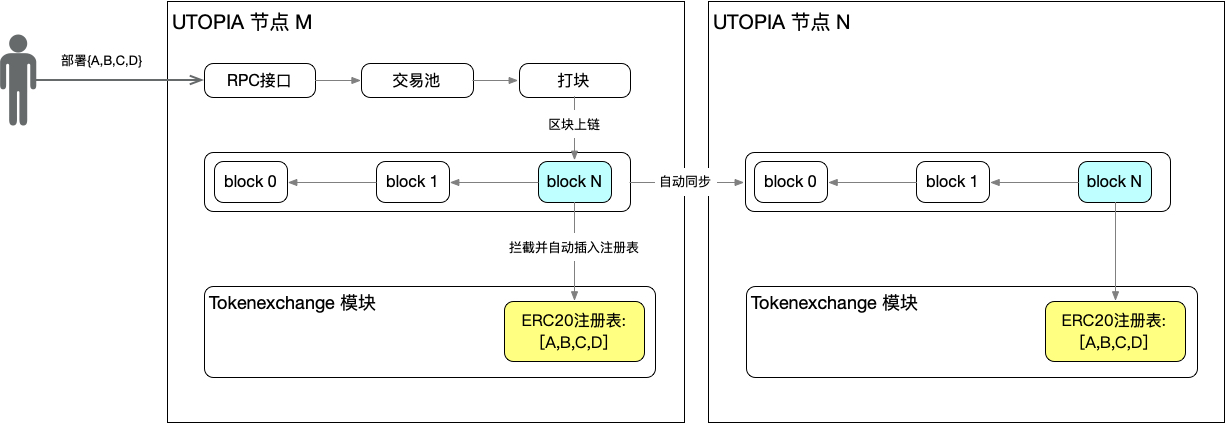
2、提供挂单功能，ASK / BID 分别对应卖单和买单，挂单时价格以交易区价格为基准，例如在 A 交易区的 B/A 交易对，挂 “ASK单” 价格为 600B/200A 时,  
表示要卖出 600B 换回 200A，挂 “BID单” 价格为 100A/300B 时，表示要用 100A 的价格买入 300B, 关于挂单价格我们不能使用小数，因为在去中心化  
合约中完成交易，合约并不支持浮点数，所以我们要以明确的兑换比例来表示价格;

3、提供按 “交易对” 归类检索挂单列表功能。例如：假设当前存在 A、B、C、D 四个交易区，想要查看 A 交易区时，提供 B/A 、C/A、D/A 三组交易对，  
进而可以通过交易对 B/A 来检索当前 B/A 的 ASK/BID 列表；

1. 提供自动撮合功能，当某组 ASK / BID 符合撮合条件时，生成撮合任务，并在下一个区块中进行撮合交易，撮合交易需要由系统自动完成，  
   买卖双方不用再次参与确认；

## 用例

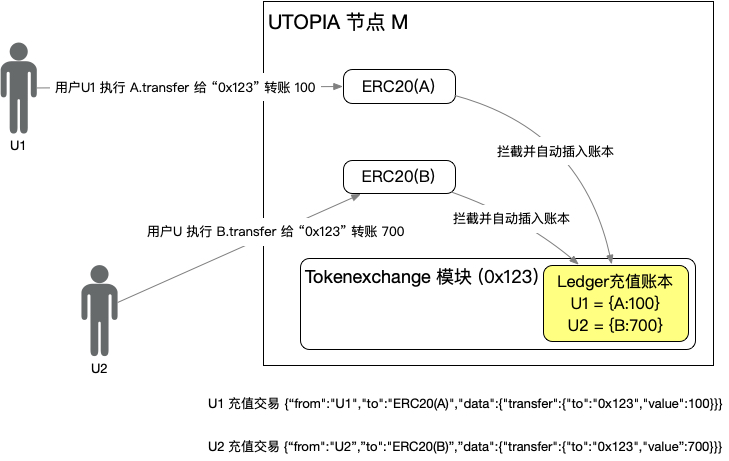
### 发布资产



启动 Tokenexchange 模块的 Utopia 主链或子链都具备自动识别 ERC20 合约并将其添加到注册表功能，可交易的 “ERC20 虚拟资产列表“ 来自此注册表，Dapp终端可以根据这个列表两两一组生成交易对，下文中会有详细描述；

如上图所示，当用户在 “节点M” 部署合约时，系统自动识别并生成了包含 [A,B,C,D] 的注册表，当区块 “block N” 从 “节点M” 同步到 “节点N” 时同样的注册表也会创建在 “节点N” 上；

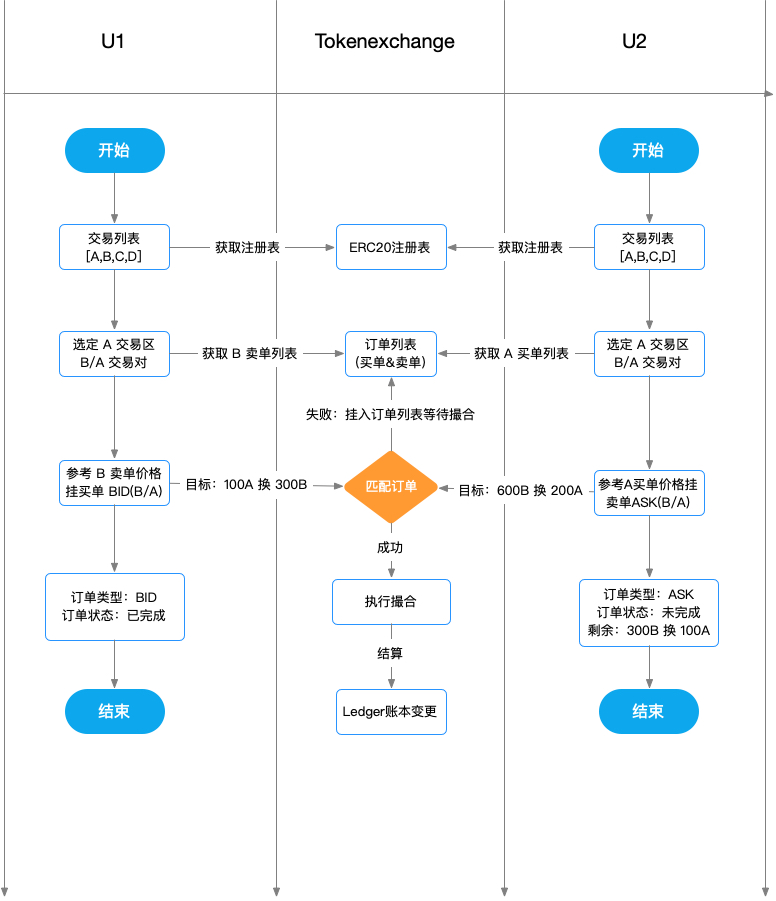
### 充值



假设用户U1 持有ERC20资产A，此时用户想通过 Tokenexchange 来交易 A，则首先需要在 A 合约上向 “0x123”（Tokenexchange 的合约地址） 地址转账，转账金额将会作为用户 U1 在 Tokenexchange 账本数据中 A 资产对应的余额；

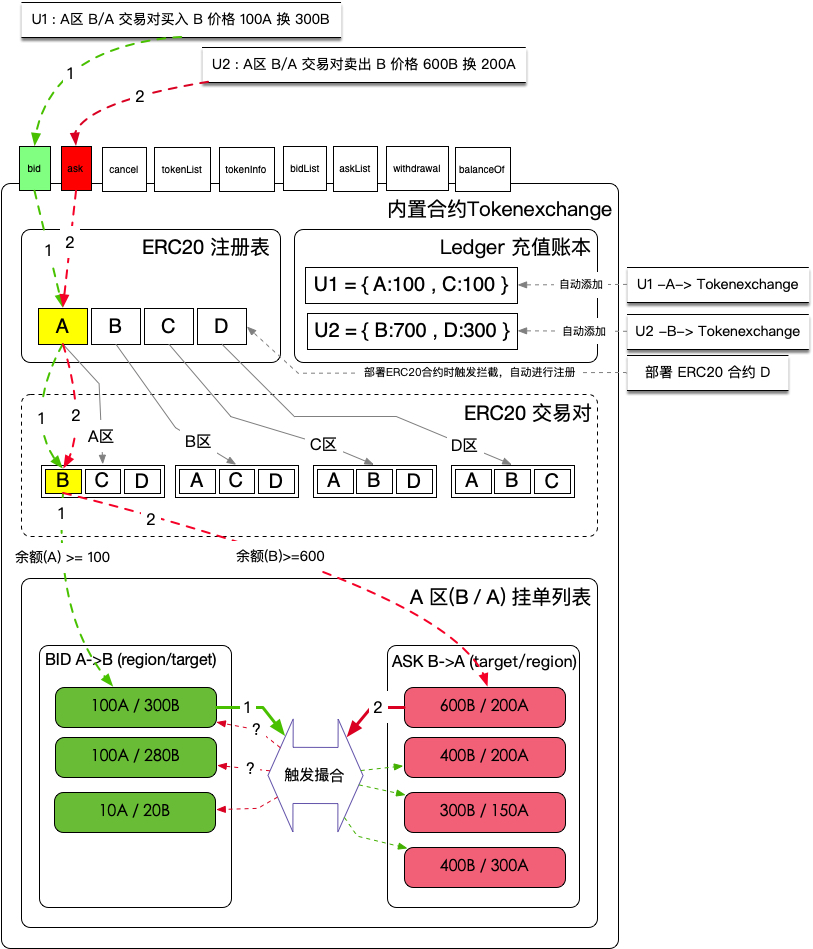
上图分别演示了 U1 充值 100A ，U2 充值 700B 到 Tokenexchange 的流程；

### 交易



当 U1 和 U2 已经完成充值并想要进行资产交换时，会遵循上图中的流程，完成最终的资产交换；

## 架构



### 结构说明

#### ERC20 注册表

运营人员部署 ERC20 合约 [A,B,C,D] 到 Utopia 时系统将自动识别并添加符合ERC20规范的合约地址到 Tokenexchange 模块的注册表；

#### **2、Ledger 充值账本**

当用户将 erc20 token 转给 tokenexchange 地址时，触发充值账本的记账功能，可以通过 balanceOf 查询余额;

#### **ERC20 交易对**

当这册表的合约数量大于等于2以后，就可以加工成交易对，注册表中的每个合约都可以单独成为一个交易区，跟其他合约的交换称作交易对，例如想要进入 A 区 A 和 B 的交易对则需要选择 “A 区”-> “B/A 交易对”；

#### **挂单列表**

上图中通过 BID / ASK 接口的调用分别在挂单列表 “BID A->B” 和 “ASK B->A” 中产生了两条挂单数据，顺序是 BID在前，ASK 在后，所以可以看到编号为 1 的 “触发撮合” 并不会产生撮合任务，而编号为 2 的 “触发撮合” 会产生两笔撮合任务，分别对应 BID 列表按价格倒序的前两条数据；

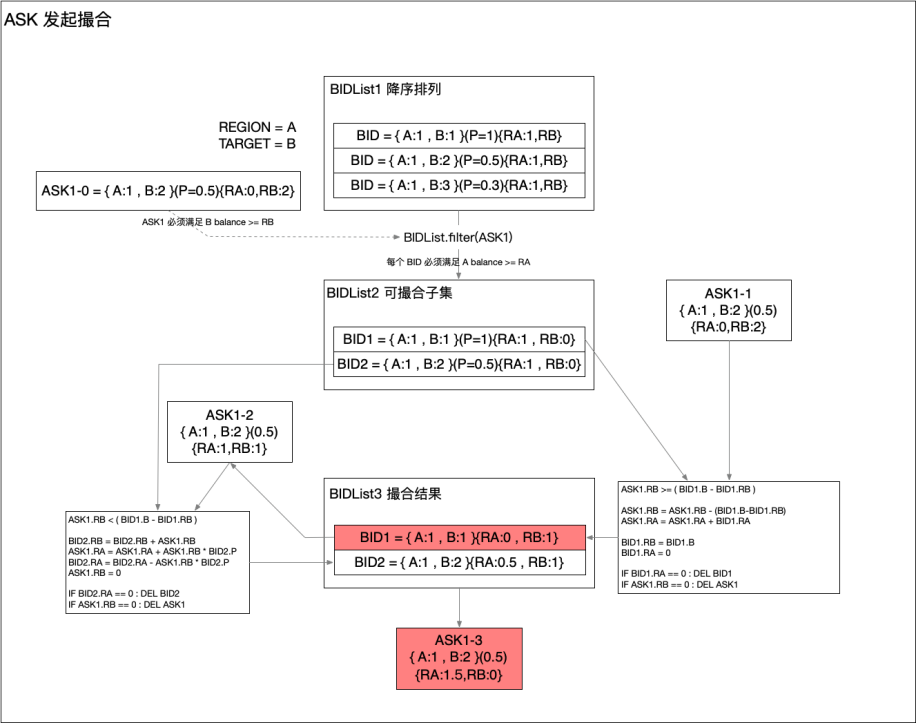
#### **触发撮合**

此处的触发撮合是产生撮合任务，当区块的打包人发现有撮合任务需要执行时，才会具体的触发撮合流程；

### 撮合算法

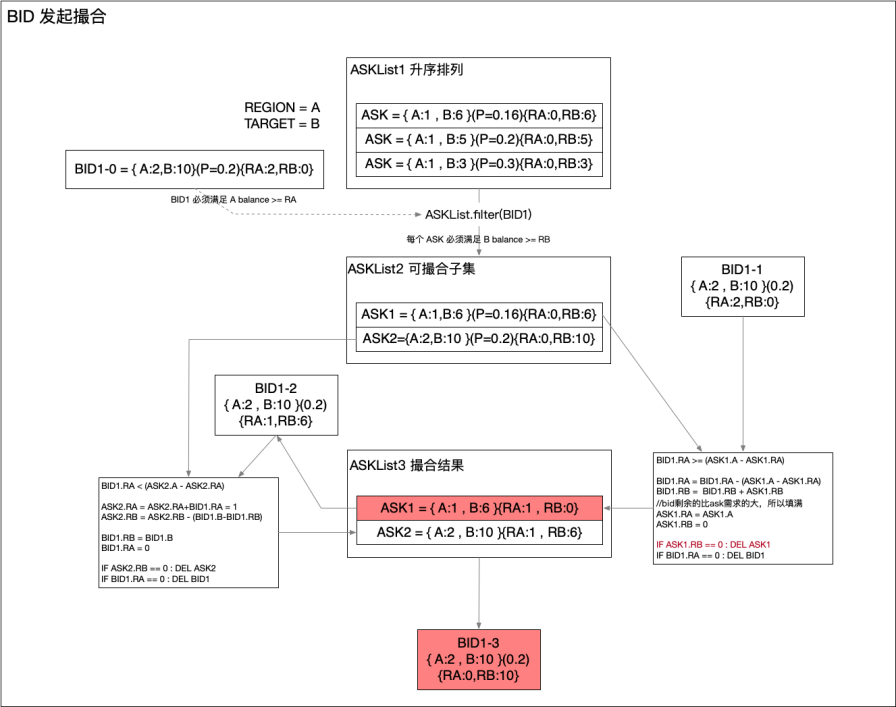
撮合在挂单的同时会自动触发，可以由 ASK / BID 单分别触发，并实时结算，具体撮合逻辑按照下面规则执行

#### 1、ASK 发起撮合



由 ASK 单触发撮合时，会过滤 BIDList 以便筛选出满足撮合条件的数据,筛选条件是 BID.Price >= ASK.Price

#### 2、BID 发起撮合



由 BID 单触发撮合时，会过滤 ASKList 以便筛选出满足撮合条件的数据,筛选条件同样是 BID.Price >= ASK.Price

#### 3、撮合原则

整个的撮合原则是价高者得，按高价成交，

例如：

买单挂 BID={A:100,B:100} 相当于价格 P = 1.0

卖单挂 ASK={A:100,B:200} 相当于价格 P = 0.5

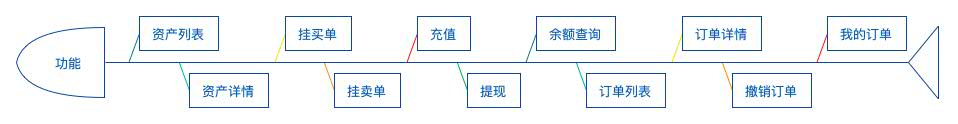
那么此时的撮合结果就是

BID={A:100,B:100}{RA:0,RB:100}

ASK={A:100,B:200}{RA:100,RB:100}

我们看到 ASK 目标是要卖 200B 换取 100A，但是按照 BID 的出价撮合完成时，ASK 得到了 100A 同时还剩余 100B

## 功能描述



### 资产列表 (tokenList)

当前链上符合 ERC20 规范的合约地址列表, 开发者可以根据列表来生成交易对

例如：

tokenlist 返回 [A,B,C,D]

以 A 为交易区，我们可以展示 B/A 、C/A、D/A 三个交易对

在接口参数中，我们会用 `address region` 表示交易区，`address target` 表示 交易目标，交易对 `B/A` 等价于 `{region:A , target:B}`；

### 资产详情 (tokenInfo)

根据 ERC20 合约地址，获取摘要信息，可以用在展示交易区和交易对时使用，详情至少会包含：资产名称、资产符号、发行量、精度 等；

### 挂买单 (Bid)

当用户持有 ERC20 资产 region，想要在region 交易区的 target/region 交易对上换取 amount 个 target 资产时，可以通过挂买单的形式，购买 target 资产

参数设计：

region : 交易区 erc20 合约地址

target : 交易目标 erc20 合约地址

regionAmount : 想要用 regionAmount 个 region token 换取 targetAmount 个 target token

targetAmount : 想要用 regionAmount 个 region token 换取 targetAmount 个 target token

### 挂卖单 (Ask)

当用户持有 ERC20 资产 target，想要在region 交易区的 target/region 交易对上换取 amount 个 region 资产时，可以通过挂卖单的形式，出售 target 资产

参数设计：

region : 交易区 erc20 合约地址

target : 交易目标 erc20 合约地址

regionAmount : 想要用 targetAmount 个 target token 换取 regionAmount 个 region token

targetAmount : 想要用 targetAmount 个 target token 换取 regionAmount 个 region token

### 充值

Tokenexchange 会以合约的形式提供一个 address 来供 Dapp 开发者和终端用户使用，那么这个地址同样用户账户充值，我们假设 Tokenexchange 合约地址为 0x123 ，用户持有的 ERC20 资产为 M，则用户需要在合约 M 上执行转账交易给 0x123 ，即可完成充值；

### 提现 (Withdrawal)

与充值相对应，用户可以通过在 Tokenexchange 合约上执行 Withdrawal 方法完成体现操作，假设用户充值的资产为 M ，则体现后剩余资产会原路返回 M 合约；

### 余额查询 (BalanceOf)

查询余额，根据 erc20 合约地址查询在 tokenexchange 中的余额，余额是通过向 tokenexchange 合约地址充值 region 对应的 token 得来的

例如：用户 U 持有 A 资产，则 U 去执行 A.transfer(tokenexchange.address,amount) 成功后，用户U再去执行tokenexchange.balanceOf(A) 时将会得到 (A.name,A,symbol,amount,decimals) 元组

### 订单列表(OrderList)

查询挂单列表，在指定的交易对上进行查询，只返回订单 ID 列表，注意不要修改列表顺序，列表已经按价格进行排序，ask 单是升序，bid 单是降序

参数设计：

orderType : 订单类型，可选值为 "bid" / "ask" 分别表示 买单 / 卖单

region : 交易区 erc20 合约地址

target : 交易目标 erc20 合约地址

### 订单详情(OrderInfo)

详情包含了挂单时的全部信息，同时还包含了订单的当前状态，其中 regionComplete / targetComplete 是当前已经撮合成的数量isFinal == true 时表示订单为最终转改，不再参与撮合，两种情况会让订单变为最终状态，一是撮合完成，二是撤单

### 撤销订单(Cancel)

订单在达到 final 状态之前均可以撤销，撤销后状态变为 final 即不再参与撮合

### 我的订单(OwnerOrder)

查询我挂过的订单 (bid & ask)

参数设计：

addrs :

是一个数组，其长度必须是 1 或 3,

是 1 时 addrs = [owner] 表示查询 owner 的全部挂单信息

是 3 时 addrs = [owner,region,target] 表示查询 owner 在 target/region 交易区的挂单信息

pageNum : 分页检索时用来表示页号，每页20条信息；