

TideAsset项目第一期设计说明书

本文是在需求分析说明书的基础上, 对项目需求进一步细化说明. 并对需求实现给出设计方案. 为项目实现提供开发依据.

1. 需求说明

本期需求包含4大核心服务:

- tAssets 稳定币服务: 参考maker协议稳定币DAI的模式, 发行我们的稳定币tsUSD.
- tLend 借贷服务: 参考Radian Capital协议, 实现跨链的借贷服务.
- tSwap AMM现货服务: 参考Thena协议, 实现AMM现货交易服务.
- tPerp AMM永续服务: 参考GMX协议, 实现AMM永续交易服务.

每种服务根据参考的协议, 需要实现我们自己的协议内容, 具体内容以下各个小节分别描述.

除了上述协议, 还要实现我们的核心资产, TDT和TCA:

- TDT: TDT主要作为各个核心服务的DAO token. 通过购买发行. 通过增发和销毁的方式维护TDT价格稳定. 通过其他核心服务的收益奖励, 实现TDT增值.
- TCA: 可以看做是另一种稳定币, 只不过tsUSD锚定的美元价格; 而TCA锚定的是一揽子代币价格, 可以作为一揽子代币的ETF. 比如: BTC, ETH, DAI, USDT, TDT 等, 通过购买发行, 初始价格是1USD, 锚定资金池的实时价格, 资金池实时价格是: 所有代币价值总和(以USD计算)/TCA总流通量. 通过设置资金池各个代币的价值比例, 尽量维持TCA价格的稳定. 通过各个核心服务的奖励, 比如定期往资金池注入TDT奖励或者部分交易手续费注入等, 实现TCA增值.

1.1 代币经济

1.1.1 TDT

- 发行总量: 2100亿
- 初始发行: 2100万, 其中100万初始化流动性, 100万市场运营, 100万社区推广, 剩余1800万用于初始化核心服务的DAO, 比如tLend, tSwap, tPerp等. 这2100万不计入发行量, 但是计入流通量.
- 发行方式: 购买发行. 初始定价为 1 USD, 通过合约购买发行 通过tSwap 添加流动性来购买发行. 购买的代币必须是TDT DAO 批准的代币. 购买的代币将存入TDT Vault(金库).

- 发行价格: 通过tSwap实时价格和外部Oracal的市场价格, 通过加权平均给出实时的发行价格.
- 销毁价格: 如果用户不在市场上出售, 那么可以选择在TDT Vault出售TDT, 出售的TDT价格是金库所有代币总价值 / TDT总发行量. TDT合约将销毁用户出售的TDT.
- 发行价格 / 销毁价格: 此数值应该在0.95--2.0之间波动, 超过2.0的时候, 合约将增发TDT, 让此数值降到1.0; 如果此数值低于0.95, 合约金库买入TDT并销毁, 直到此数值升到1.0.
- 增发: 当发行量大于 $21,000,000 * 20 = 420,000,000$ (4.2亿, 加上之前的2100万)时, 增发5%, 比如用户购买100万TDT, 实际发行105万, 其中100万给用户, 4万给项目方, 1万给到DAO奖励.
- DAO token: TDT的主要作用是协议管理DAO token. TDT是TDT和TCA合约DAO token, 还是tAssets协议的DAO token, 当锁定TDT时, 成为veTDT token, 这时1个veTDT代表一票的投票权重. 同时, 在tLend, tSwap, tPerp等协议启动时, 作为这三个协议的初始化DAO token.
- 增值:
 - 核心服务的部分奖励(tLend, tSwap, tPerp等代币排放和交易手续费等)经进入到TDT金库.
 - 金库资产投资: 到tLend提供借贷, 赚取存款利息; 到tSwap和tPerp中提供流动性, 赚取流动性奖励.
- 没有抵押物的概念, 只有金库资产的概念. 这些资产需要在 DAO 的白名单里. 每种资产需要设置上限比例和下限比例, 也就是此资产在所有资产中的占比(以美元计).
- 用户买卖TDT有两种方式:
 - 在TDT金库买卖
 - 在tSwap买卖
- 开始运行时: 白名单token和占比(min, max)如下(这里只是为了说明, 具体多少在上线时确定):
 - USDT: (10, 30)
 - USDC: (10, 30)
 - DAI: (10, 30)
 - BTC: (5, 10)
 - ETH: (5, 10)
 - TCAV1: (5, 10)

- TCAV2: (5, 10)
- TTL: (1, 5)
- TTS: (1, 5)
- TTP: (1, 5)
- 如果某种token低于最小值或高于最大值, 用户购买的时候, 需要缴纳手续费, 手续费的比例和超额比例成正比. 比如现在USDT已经到达30%, 假如金库总价值是1亿, USDT是3000万, 如果你用USDT购买100万TDT, 那么:

手续费为: $100\text{万} * (3100\text{万}/1.01\text{亿} - 30\%) / 10$. 手续费是700 USDT.

- 用户可以到tSwap兑换其他的代币购买TDT, 避免购买时产生手续费.
- TDT 质押锁定后生成veTDT, 最长周期为4年, 抵押年限越长, 收益越大. 1个TDT, 比如给出的是按1.025基准月复利计算: 抵押一个月, 获得1.025个veTDT, 那么6个月为 $(1.025^{**} 6)$, 1年为 $(1.025^{**} 12)$ 依次类推:
 - 1个月: **1.025**个veTDT
 - 6个月: **1.159563**个veTDT
 - 1年: **1.345541**个veTDT
 - 2年: **1.811364**个veTDT
 - 4年: **3.281072**veTDT
- 锁定后, 只有到期才能解锁, 中途无法解锁.(这条有必要的的原因是, 用户选择质押后, 投票的权重就是固定的, 比如, 1个TTL质押4年, 那么当前的权重是3.28. 如果允许用户解除质押, 那么)
- TDT 金库将会拿出部分资产到tSwap, tLend, tPerp(暂时三种, 以后增加外部的Dex)做市, 以增加流动性, 赚取流动性奖励以及交易手续费和借贷手续费.
 - 每种资产(token)投资到tLend, tSwap, tPerp的比例都不同, 在DAO中设定.
 - 如果获取的奖励token不在TDT金库的资产白名单里, 那么这些奖励放入到TDT dao中.
 - 奖励暂存在对应的服务金库里, 如果获取需要调用TDT claim接口, 比如tLendClaim, tSwapClaim, tPerpClaim等. 这些可以由任何人发起操作.
 - 这个类似于基金的操作方式, 操作者为基金经理的角色, 前期基金经理的角色由DAO的管理人员负责, 投资买入和卖出. 后期由用户负责(没有风险的时候).

1.1.2 TCA V1: Stable ETF

- TCA和TDT 差不多. 不同之处在于:
 - TDT作为tAssets的DAO token(抵押TDT, 产生veTDT). 管理tAssets服务参数, 包括TDT, TCA和tsUSD等参数.
 - TCA发行没有限制. 发行多少由用户的购买意愿决定.
- 发行方式: 购买发行. **通过合约购买发行**, 在tSwap上购买发行. 购买代币必须是稳定币, 比如USDT, USDC, DAI等, 必须是合约DAO白名单批准的. 购买的稳定币代币进入TCA 金库.
- 发行价格: 初始价格是 1 USD, 然后根据金库中价值, 计算发行价格: 总价值 / 发行数量
- 销毁: 用户可以选择在金库兑换 TCA, 价格按发行价格计算, (是否需要支付一些费用?), 金库将销毁掉收到的TCA.
- 增值: 和TDT一样.
- 当投资以及其他获取的奖励, 如果不在资产白名单中, 那么将这些奖励存入DAO中.

1.1.3 TCA V2: Vault ETF

TCA V2是在TCA V1的基础上增加其他代币, 形成一种一揽子代币的ETF(指数基金). 发行和销毁的价格, 也是按照金库的实时价格计算得出.

1.1.4 tsUSD

tsUSD是我们发行的稳定币. 价格锚定1美元. 实现方式, 在第一期中, 和maker协议DAI一样. 超额抵押发行.

也可以在tSwap购买现货.

以后可能有tsHK.

1.1.5 协议代币

tAssets, tLend, tSwap和tPerp 核心协议, 每个协议都有协议代币:

- tAssets: TDT
 - TDT在DAO抵押锁定后, 成为veTDT, 对应maker协议的MKR代币, 类似MKR 管理代币的功能
 - TDT过多, 我们可能要设置抵押锁定的veTDT数量, 比如100万.

- MKR还有资产重组的功能, 即Dai债务无法偿还时, 发行MKR偿还购买Dai偿还. 盈余拍卖后, 销毁MKR. TDT实现MKR资产重组的功能.
- tLend: TTL, 对应Radiant 的RDNT 代币. 抵押锁定TTL, 产生veTTL, 用于DAO管理.
- tSwap: TTS, 对应Thena的THE 代币. 抵押锁定TTS, 产生veTTS, 对应Thena的veTHE代币, 用于对lp pool投票. 我们暂不实现theNFT?.
- tPerp: TTP, 对应GMX的GMX 代币. TTP锁定抵押为veTTP, 用于DAO管理.
 - GMX通过质押GMX代币, 产生三种奖励: esGMX, Multiplier Point以及ETH/AVAX 等奖励
 - esGMX可以和GMX一样质押, 产生和GMX一样的奖励
 - esGMX还可以通过vest, 一年内转换为等量的GMX.
 - 我们可以仿照以上实现, 质押TTP, 产生esTTP, vest esTTP, 兑换为TTP.

协议代币的发行规则如下:

- 代币总量: 1亿(100,000,000) +
- 初始发行: 1亿, 直接放到下面4个锁仓合约:
 - tsaDAO: 10,000,000: 50个月线性解锁
 - DAO: 10,000,000: 初始化解锁2,000,000, IDO发行; 剩余8,000,000, 80个月线性解锁, 每个月解锁100,000;
 - Dev: 100,000,000: 先锁定10个月, 然后20个月线性解锁;
 - LP/veToken staking奖励: 700,000,000: 下个月(周)解锁多少, 由DAO提案投票决定.

1.1.6 DAO token

我们主要有4种DAO token, 实现不同协议的治理. 在第一期中, 每种协议有各自的DAO token, 后续会把veTDT 作为通用的DAO token, 占据部分权重, 比如50%, 和协议各自的DAO token一起治理协议.

第一期里, 每种协议DAO token如下:

- tAssets: veTDT, 对TDT锁定后, 形成veTDT. 对应maker协议的MKR. tAssets协议负责发行TDT, TCA和tsUSD. veTDT则负责治理tAssets协议. 主要的系统参数的调整.
- tLend: veTTL, 对应Radiant的RDNT.
- tSwap: veTTS, Thena没有实现gov内容, veTTS对应veTHE.
- tPerp: veTTP, 对应GMX的GMX(esGMX)

以上DAO token, 不能转移给其他人, 只能到期解锁后, 变为相应的token, 才可以转移.

持有DAO 代币会获取如下奖励:

- TDT增发: TDT发行 > 4.2亿的时候, 会增加5%, 其中1%用于DAO token的奖励.
- 协议代币排放奖励: 每个协议排放奖励不同, veTDT 没有协议代币排放, 可能在TDT增加奖励比例会比较高. TCA 收益转为TDT DAO. 如何发放奖励.
- 协议费用奖励:
 - veTTL 获得一定的借贷费用
 - veTTS 获取一定的交易费用
 - veTTP 获取一定的交易费用

1.1.7 流动性代币

1. Radiant

- 用户存款和取款都有利息, 和银行类似. Radiant收益一部分来自利息差.
- 如果用户只存款没有锁定dLP, 那么用户只能获得存款利率(APY), 没有RDNT的排放奖励.
- 如果用户存款,并且有至少5%的dLP锁定, 那么用户除了APY, 还有RDNT排放的奖励
- dLP 有两种获取方式:
 - Arbitrum: Balancer 80/20 composition (80% RDNT & 20% ETH)
 - BNB Chain: Pancakeswap 50/50 (50% RDNT & 50% BNB)

Balancer和Pancakeswap是两个去中心化的交易所, 增加流动性获得lp token.这里的80/20, 50/50是添加流动性时的资产价值的比例, 不是数量.

- 因为Radiant分别部署在Arb和BNB链上, 所以dLP token不同, 收益也不同. 用户可以根据收益自己挑选.
- 用户使用zap功能, 直接把自己的资产(ETH, BNB和RDNT, 如果没有ETH等, 可以借出来), 直接抵押到Blancer和Pancakeswap 相应的lp 池子里, 换取lp token, 然后把lp token抵押到(stake) 系统里, 这样比自己操作少3个步骤(3个交易)

2. Thena

- Thena每个lp pool都有对应的lp token;

- 用户添加流动性到lp pool会获得相应的lp token;
- 用户锁定lp token会获得THE排放奖励(添加流动性比例必须是50:50)
- 持有veTHE的用户每周会对 lp pool投票;
- 根据票数, lp pool会获取THE排放奖励;
- 锁定lp token的用户会根据权重获得相应的THE排放奖励.
- 添加流动性的用户会获取交易费的奖励, 不同类型的池子, 奖励不同.

可以添加单币种, 和 杠杆挖矿(借和抵押tLend).

3. GMX

- 我们主要关注GMX V2: GM代替GLP. 通过单独的池子提供流动性. GLP通过9种代币构成指数(每种代币比例不同, ETH, BTC和USD比例超过90%).
- 流动性提供者可以从杠杆交易、借贷费和兑换中赚取费用。
- 流动性提供者可以通过购买或出售GM代币来加入或退出GM池。
- GM代币的价格取决于长仓代币和空仓代币的价格以及交易者开仓的净盈亏。交易费用会自动提高GM代币的价格。
- 也就是每个池子的GM代币价格不同. 每种GM代币只能在对应的池子里买和卖(应该是deposit和withdraw更准确, 代码里是这两个函数, 但是由于GM代币价格会随着时间改变, 所以买卖混存在价差)
- 每个池子, 也叫Market, 有4种代币组成:
 - GMX Market Token(GM): 用户Deposit Long/Short中的一种或者两种token, 换取的代币.
 - Index Token: 指数代币, 做空做多的指数
 - Long Token: 多头抵押代币
 - Short Token: 空头抵押代币
- 购买GM 就意味着给池子添加流动性.
- 持有GM代币没有GMX排放奖励, 奖励是交易费和借贷费.
- GM的持有者实际是用户交易对手方, 如果用户获利, 意味着GM持有人亏损.

4. tLend, tSwap, tPerp的设计:

- 我们保留以上实现
- tLend中, 我们根据部署的平台, dLP获取方式不同. dLP将获得借贷费用奖励和TTL排放奖励.
- tSwap中, 和Thena一致就好. 抵押 lp token, 会获得TTS排放奖励和交易费用奖励.
- tPerp中, 我们和GMX V2 一致. 持有lp token(market token)将获得交易费用加奖励和TTP排放奖励.

1.2 IDO

- 类似pancakeSwap IFO 功能
- 发行用户需要有TDT, 作为IDO 手续费
- 发行用户需要质押TDT/TCA, 提供流动性, 才能开启IDO.
- 初始价格为 1 USD(其他代币需要用户设定价格), 但是需要使用TDT购买, 比如当前TDT价格是2 USD, 那么发行价格为0.5TDT.
- 参与用户必须质押的 TDT/TCA lp 代币, 才能申请购买等价值的IDO 代币.
- 或者参与用户使用已有的资产, 兑换(swap)为TDT/TCA, 并质押锁仓. (同上)
- 参与用户使用TDT购买, 以及白名单资产兑换为TDT购买.
- 销售结束后, 参与用户获得相应的 IDO 代币. 发行用户获取TDT代币.
- 如果是超额认购, 按比例发放代币, 并可取回多余的TDT.
- 认购结束后, 发行用户需要锁仓TDT, 比如和发行的TTL/TDT 添加流动性池, 锁仓6个月.

1.3 tAssets - 稳定币

maker协议DAI的功能, 为方便跨链, DAI实现为layerZero OFT方式.

TDT和TCA的实现.

1.4 tLend - 跨链借贷

- 跨链借贷指的是, 用户在A链抵押资产, 可以在所有ETH兼容链里借出资产.
- dLP
- zap

- loop

1.5 tSwap - AMM 现货DEX

- swap
- Limit
- twap
- cross-chain swap
- gauge
- bribe
- lp pool
 - fusion
 - 添加流动性: 支持单币种.
- 池子里的部分资金 50% 可以放到 tLend 中去放贷, 获取存款利息(具体多少需要DAO管理)
- 当某个池子的资产低于某个阈值的时候, 可以从tLend取回.

1.6 tPerp - AMM 永续DEX

- swap
- trad:
 - long/short
 - swap
- buy
- 池子里的部分资金 50% 可以放到 tLend 中去放贷, 获取存款利息
- 当池子的某种资产低于某个阈值的时候, 可以从tLend中取出来.
- 对冲 TDT和TCA: 当在tSwap 购买或者出售TDT, TCA时, 为避免价格波动过大, 可以在tPerp 对TDT和TCA追加头寸进行对冲. (由用户选择是否做对冲, 比如使用多少资金占比, 以及头寸价格等)

1.7 Oracle - 价格, 随机数, 区块头等预言机

1.7.1 现有协议Oracle的使用

1. Maker:

Maker 协议通过Orcacle 获取抵押物的价格, 以确保用户的仓位有足够的抵押率.

- OSM 合约负责Oracle 的价格安全, 把价格延迟1小时再提交给系统. 为了检测价格的合理性和应对Oracle攻击. OSM 读取Median合约获得价格信息. 同时, 有白名单机制, 也就是经过DAO批准的Oracle才能报价, 否则无效.
- Median 合约负责计算白名单价格的中值. 同时也维护白名单. OSM以及Median是1:1的.
- [omnia-relay](#) 是Oracle客户端工具, 它使用安全的 scuttlebutt 网络来传递离线的价格数据, 并在链上验证身份和真实性. Omnia relay 有两个主要模块: Feed 和 Relay. Feed 模块从不同的数据源获取价格, 并用以太坊私钥签名. Relay 模块监控广播的消息, 检查活跃度, 并将价格数据和签名整合成一个以太坊交易, 提交到链上的 Oracle 合约.
- Feed 从Gofer模块获取签名的价格信息. Gofer从配置的数据源获取价格信息.
- relay 把价格信息发送给 Median合约

2. Radiant

Radiant V2里使用Chainlink Oracle 喂价. 具体在两个模块中使用:

- lending: 核心业务借贷模块, 这个模块的源码是基于aave的. 使用Oracle主要是在AaveOracle合约里:

```
mapping(address => IChainlinkAggregator) private assetsSources;
```

这个映射包含了借贷业务中market各种资产的价格源, IChainlinkAggregator 是聚合器代理合约 EACAggregatorProxy 的接口. EACAggregatorProxy合约也就是真正的数据源.

- radian: 这是Radian实现自己业务的模块, 比如dLP. 在子模块 oracle里, 合约 PriceProvider就是提供dLP价格的合约. baseTokenPriceInUsdProxyAggregator在arb链表示ETH价格的oracle(IChainlinkAggregator->EACAggregatorProxy). 其中

```
IBaseOracle public oracle;
```

这个oracle是用来获取RDNT价格, 合约实例为[ChainlinkV3Adapter]
([ChainlinkV3Adapter.sol](#) — [WORKSPACE](#) — [Blockscan contract source code viewer](#)).

3. Thena

无

4. GMX

在V2版本中, 实现了自己的Oracle, 也就是可以向外报价.

1.7.2 我们的要求

TDT, TCA以及tsUSD 中金库的抵押物价格需要Oracle和tSwap共同报价. 所以tSwap需要实现Oracle接口(参考GMX V2 oracle模块, 或者简单实现类似`getPrice(address token) return uint256`的接口给我们自己开放). tsUSD Oracle参考Maker协议的实现, 使用[omnia-relay](#) 为资产报价. TDT和TCA的外部价格使用Chainlink, 实现参考Radiant.

tLend 用户存入资产和借出资产的价格需要Oracle和tSwap共同报价. 这里需要Oracle实现参考Radiant的实现, 使用Chainlink.

1.8 Bridge - 跨链桥, 不同区块链之间的资产转移和消息通信

我们的项目需要同时在多个链上运行, 我们的TDT, TCA和tsUSD稳定币需要再多个链上转移. 我们的tLend, tSwap和tPerp也需要同时部署到多个链上, 初步计划在eth, op(Optimism), arb(Arbitrum)和base上, 进一步视难度上 bsc/matic/avax 等平行链和 zksync/linear 等还没起来的L2。

对应的协议当前多链情况如下:

- maker: 在eth 上部署发行, 其他所有L2以及BNB等都有相应的DAI, 一般都是通过自由的桥来实现跨链转移Dai的.
- Radiant: 同时部署在Arb和bnb(bsc)上. 采用layerZero跨链方案, 实现全链借贷(EVM兼容), RDNT代币采用layerZero OFT格式, 将使跨链手续费共享更加无缝, 使在其他链上更快地启动, 并允许对桥接合约进行本地所有权, 而不是依赖于第三方桥接.
 - [StargateBorrow]([StargateBorrow.sol — WORKSPACE — Blockscan contract source code viewer](#)) 合约实现跨链借贷. 包含了[router]([Router.sol — WORKSPACE — Blockscan contract source code viewer](#)), 负责实现跨链交换swap.
- Thena: 只在bnb上部署, 但是通过使用squidrouter+Axelar跨链技术, 可以在实现不同链跨链swap.
- GMX: 同时部署在arb和avax(Avalanche)上, 没看到关于跨链的实现, 用户不能实现跨链交易之类的操作. 但是GMX, GLP和GM(market token), 通过钱包可以实现跨链转移.

我们的实现要求:

- 4种服务在eth, arb, op和base上同时部署.
- 代币使用layerZero OFTV2形式, 可以方便在不同链之间转移资产.

跨链桥是一种链与链连接的桥梁工具, 允许将代币、资产、数据从一条链转移到另一条链。两条链可以有不同的协议、规则和治理模型, 而桥提供了一种相互通信和兼容的方式来安全地在双方进行互操作。

LayerZero 是一种全链互操作性协议, 专注于链与链之间的数据消息传递。在业内, 目前有一种说法将此类的“桥”称为: “Arbitrary Messaging Bridges (AMBs)”, 即任意信息传递桥, 这些桥允许任何数据, 包括代币、链的状态、合约调用、NFT 或治理投票等, 从链A转移到链B。

LayerZero 最突出的特点是其超轻量级的节点, 利用超轻节点技术, 通过中继者和预言机在不同链的端点之间传输消息, 在保证安全性的前提下降低费用。

我们需要自己提供预言机和中继者, 而且保持独立, 保证不串通, 来保证跨链的安全。

1.9 治理

1.9.1 参考协议的治理

我们从治理内容和治理方式来描述我们参考的现有协议的治理过程. 然后总结我们的治理内容。

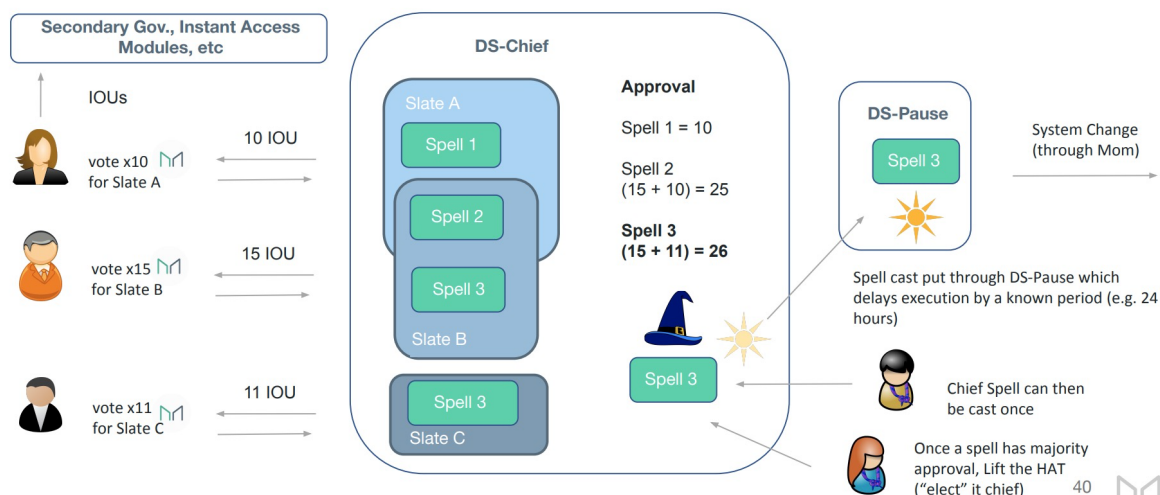
1.9.1.1 Maker 协议的治理

- 治理内容: 主要是金库的风险参数
 - **债务上限 (Debt Ceiling)**: 债务上限指的是一种担保品所能生成出债务总额上限。Maker 治理为每一种担保物都设定了债务上限, 以确保 Maker 协议的担保物组合具备足够的多样性。一旦某种担保物达到了债务上限, 就不可能产生更多债务, 除非已有用户偿还部分或全部的金库债务 (从而释放出债务空间) 。
 - **稳定费 (Stability Fee)**: 稳定费是根据一个金库所生成的 Dai 数量来计算的年利息 (对生成 Dai 的用户来说, 稳定费率相当于贷款的年化利率; 对 Maker 协议来说, 稳定费率相当于年化的收益率)。稳定费只能由 Dai 支付, 发送至 Maker 缓冲金 。
 - **清算率 (Liquidation Ratio)**: 清算率较低, 意味着 Maker 治理对担保物价格波动性的预期较低; 清算率较高, 意味着对价格波动性的预期较高。
 - **清算罚金 (Liquidation Penalty)**: 清算罚金是当清算发生时, 根据金库中未偿还 Dai 的总量向用户收取的额外一笔费用。清算罚金旨在鼓励金库所有者将质押率保持在适当的水平。

- **担保物拍卖期 (Collateral Auction Duration)** : 每种 Maker 金库的担保物拍卖时间上限都是 特定的。债务拍卖期和盈余拍卖期则是系统全局参数。
- **竞拍期 (Auction Bid Duration)** : 单次竞拍结束及终结之前的最低时长。
- **最低加价幅度 (Auction Step Size)** : 这一风险参数旨在激励拍卖中的早期竞拍者, 防止加价幅度过低的情况泛滥。
- **治理过程:**
 - **提议投票:** 提议投票的目的是在进行执行投票之前, 先在社区内形成一个大致的共识。这有助于确保治理决策是经过仔细考虑, 且在进入投票流程之前就已达成一致的。
 - **执行投票:** 执行投票的目的是 批准/驳回 对系统状态的更改, 例如, 投票决定新引入担保物的风险参数。
 - **具体过程:** Maker 论坛的 Signal 议案发起流程, 任何论坛参与者都可以发起 Signal 议案论坛投票并且参与投票, 在论坛内部达成一致的 Signal 议案会被 Maker 临时风险团队采纳, 并在链上发起 Governance Poll 和 Executive Vote, 如果全部通过, 相关议案就会成为 Maker 协议的一部分。其中 Governance Poll 就是提议投票, Executive Vote 就是执行投票. 也就是提议投票通过, 就会发起执行投票。
 - **Maker 临时风险团队 (Maker Foundation Interim Risk Team)** : 约 3 人, 他们的任务是管理抵押资产组合并建议适当的货币政策。目前每周定期发起的货币政策 Governance Poll 就由临时风险团队负责。
 - **Maker 治理和社区团队 (Maker Governance and Community Team)** : 约 3 人, 设计 Maker 的治理规则, 召集每周的 Governance Call, 管理 Maker 论坛和其他社区平台。

Governance Module - what

3



1.9.1.2 Radiant 协议的治理

- 治理内容: 不受限制, 可更改升级的都可以
- 治理过程: 使用snapshot 投票.

snapshot是中心化投票工具，设定好投票token的合约地址以及投票周期。那么snapshot在投票截止日期的时候，查看用户的投票token balance，就获得了用户的投票权重，能够计算出赞成，反对和弃权的比重。这样就完成了投票。完成后会把投票结果放到IPFS上存储，作为一个snapshot。

在snapshot之前, 用户会在[论坛](#)先讨论, 然后由core成员在snapshot发起投票.

1.9.1.3 Thena 协议的治理

无

1.9.1.4 GMX 协议的治理

和Radiant类似. 首先在 [GMX](#) 论坛发起讨论, 然后到snapshot上投票.

1.9.2 我们的治理

我们将采用统一的治理模型:

- 使用snapshot投票方式
- 每周投票: 周三20:00开始, 到下个周三20:00截止

3. 首先在论坛中发起讨论, 然后由管理员(票选)提交提案并发起投票
4. 超过投票权重的50%赞成票, 视为提案有效.
5. 有效提案将由管理员执行(最好有个合约来执行, 合约能够检查执行函数是否按照提案执行的. 或者采取多签方式, 比如三个管理员统一签署才有效, 这样防止某个管理员作弊或者出错)
6. 投票的来源: veTDT + veXXX(veTTL, veTTS, veTTP等). veTDT和veXXX投票权重相同.
7. 项目开始运行时, 通过veTDT启动DAO.
8. 管理员拥有紧急关停的权限, 当系统遭受攻击而无法正常运转的时候. (???是管理员还是项目方? 还是说项目方充当一名管理员?)

1.10 风险控制

1.10.1 紧急关停

tAssets基于maker 协议, 所以继承maker协议紧急关停的功能.

tLend, tSwap和tPerp也都继承对应的协议, 对此功能本期不做修改.

1.10.2 金库转账最大限度

如果遭受最严重的攻击, 攻击者会把金库里所有资产转移. 为避免这种情况发生, 当出现大额(金库资产的20%)资产转移或者短时间内, 向系统外转移金额过大(金库资产的20%), 那么系统将暂停或者暂停向外转移资产一段时间(比如24小时). 这样即使收到最严厉的攻击, 那么系统最多损失20%的资产.

2 合约设计

2.1 tAssets

这个协议主要实现3个功能:

- TDT
- TCA
- tsUSD

2.1.1 Token合约

由于我们需要在多个链上部署, 以及全链上资产转移(tLend可以支持全链资产借贷). 跨链我们是用layerZero 技术. 所以TDT, TCA, tsUSD以及TTL, TTS和TTP等token, 使用layerZero 的OFTV2方式: [合约代码](#)

合约说明:

- 继承了OFTV2, 实现跨链资产转移.
- 构造函数需要提供 layerZero endpoint 节点地址, 用来跨链
- 提供mint和burn, 用于发行和销毁. 对于TDT, TCA, tsUSD, 发行由金库合约Val控制. TTL, TTS, TTP等发行由Lock合约控制, 同时也由Vex合约控制(比如抵押TTL, 产生veTTL).
- 增加了auth和Pausable修饰符, 管理发行和终止合约运行.

2.1.2 Val合约

[合约代码](#)

合约说明:

- TDT, TCA(V1, V2)金库的管理(TDT和TCA V1, TCAV2有各自的金库)
- buy, sell函数用来管理代币core(TDT, TCA)的发行和销毁. 这里面调用了Token的mint和burn, 所以在部署是, 需要把Token的auth给到Val.
- deposit和withdraw接口: 用于金库的资产去投资. 调用InvLike接口(代表tLend, tSwap和tPerp)的deposit和withdraw.
- 设置金库资产 setAsset以及移除资产 removeAsset
- buyFee以及sellFee用来计算当超出资产限制的时候, 买卖时, 收取的费用.
- 设置资产投资最大限制setInv
- 一些view函数
- auth修饰符用于管理接口权限.
- stop 函数用于停止合约.

2.1.3 Lock合约: TTL, TTS, TTP代币锁定发行

[合约代码](#)

合约说明:

- 管理TTL, TTS和TTP等代币的发行. 参考1.1.5代币发行的说明.
- 根据remain和minted计算解锁余额
- 在_mint函数里调用了token.mint, 需要Lock合约有token的auth权限.
- 在构造函数里, 设置了1.1.5有关的参数.

2.1.4 tsUSD

我们自己的稳定币tsUSD, 采用Dai的方式, 超额抵押发行. 在第一期中, 我们在使用我们自己的治理模块, 代替maker协议MKR相关的部分.

2.2 tLend

我们基于Radiant Capital V2的代码开发. 代币包括TTL和veTTL. TTL使用Lock合约发行, veTTL使用Vex合约发行. veTTL用于Governance投票.

- 我们关注Radiant 实现的亮点功能: 跨链借贷. 是通过Stargate实现的. [stargateRouter代码](#), [stargateBorrow代码](#) (Radiant代码质量一般, 起码注释就错误)
- 和TDT, TCA金库适配, 让金库的资产可以进入tLend进行投资获取收益:
 - dLP流动性
 - Deposit借贷

2.3 tSwap

我们基于Thena协议的合约代码开发. 代币包括TTS和veTTS. TTL使用Lock合约发行, veTTS使用Vex合约发行. veTTS用于Governance投票.

- IDO: 基于pancakeswap IFO开发.
- 要和TDT, TCA等金库合约做适配, 能够获取tSwap价格, 以及能够添加流动性(deposit)获取奖励;

2.4 tPerp

我们基于GMX V2协议的合约代码开发. 代币经济部分改为自己的TTP和veTTP. TTP使用Lock合约发行, veTTP使用Vex合约发行. veTTP用于Governance投票.

- 和TDT, TCA金库合约做适配, 金库资产可以提供流动性获取奖励.

2.5 合约测试

由于我们第一期上线4个网络: eth, arb, op和base. 所以我们测试选择 eth Goerli 测试网络.

测试的目的如下:

- 找出错误;
- 争取全覆盖;
- 优化gas;
- 接口易用;

3 前端

3.1 网站结构

顶端导航, 依次是Home, tAssets, tLend, tSwap, tPerp.

3.2 Home: Dashboard

- TDT和TCA的overview
- 整体网站的介绍
- 4个核心服务的介绍: tAssets, tLend, tSwap, tPerp

3.3 tAssets

使用spark-interface 和 oasis-borrow前端为基础开发.

- tsUSD overview
- borrow : tLend borrow
- buy :
 - tSwap buy
 - Vault buy
- sell :
 - to tSwap
 - to Vault

- Governance

3.4 tLend

由于Radian Capital前端没有开源, 我们以aave前端为基础开发, 功能如下:

- your dashborad
 - lending
 - rewards
 - deposits
 - borrows
 - zip
- market
 - stats voerview
 - assets list
- bridge TTL ?
- buy dLP
- governance

3.5 tSwap

由于Thena前段没有开源, 我们采用pancakeswap前端为基础开发, 功能如下:

- swap
- liquidity
- lock
- vote
- gauges
- rewards

3.6 tPerp

我们使用GMX前端为基础进行开发, 功能如下:

- your dashborad
- buy market token
- earn
- governance
- trade