基于以太坊 layer2 Arbitrum rollup 的去中心化的组合资产衍生品交易所 _V1

Opswap 是基于以太坊二层开发的去中心化组合资产衍生品交易平台，首家采用以太坊二层扩容技术 Arbitrum Rollup 开发，基于 AMM 自动做市技术的现货数字货币交易和首家独创基于现货保证金交易的永续合约平台（不对赌）。最高可以支持 10 倍杠杆。

1.开发背景

从 2020 年六月开始，以太坊上的 DeFi 火热，以 Compound 为代表的老牌借贷协议在 Uniswap 上，将自家代币以流动性挖矿的形式上线，彻底引爆了 DeFi 领域的流动性挖矿，导致以太坊上网络异常拥堵，动辄几十美金的 GAS 费，彻底把小额玩家拒之门外。每一笔交易、每一个操作都需要等待至少一个区块确认，实时性差。暴露了以太坊扩容问题迫在眉捷 ，也是以太坊现阶段发展的一个瓶颈。

以太坊创始人 Vitalik 提出应对 2.0 推出前中期乃至长期的扩容性需求，整个以太坊生态系统需要将发力点集中到 rollup 侧链上，目前以支付方向的 ZK rollup（零知识证明）和复杂型智能合约方向 optimistic rollup（欺诈证明）的二层技术成为主流，可以预见 2021 年将是 rollup 大量项目暴发的一年。

对于 Rollup 来说，其是通过链上链下相结合的方式提高链上效率，把简单的 merkle 树放在链上，复杂的运算过程放在链下。因为目前的实践看来 Layer2 相较 Layer1 在 TPS 速度方面将有 100 倍以上的提升。而 Opswap 以 Arbitrum rollup (也叫复杂型 Optimistic Rollup)技术为基础搭建 DEX，则使得 Opswap 有着明显的优势，几乎 0 GAS 费，交易秒确认，杠杆合约交易 0 滑点，在性能和安全性方面已经全面超越中心化交易所。

2. Opswap 基于的 Arbitrum rollup 技术和其他 Rollup 解决方案的区别

Arbitrum Rollup 是由 offchain labs 团队开发的基于以太坊上的 optimistic rollup 扩容技术，而对于当下区块链技术没有一个权威的分类，所以我们就可以认为 Rollup 本质上是一种 Layer2。整体而言 Rollup 的特点在于链下的计算+链上的数据并以 Fraud proof 欺诈证明的形式验证，也就是说少量数据根我们是放到链上的，而计算数据的过程我们拿到链下来计算，很多计算十分复杂是会降低区块链系统的效率。

同时 Rollup 还不是所有数据都在链上，它的链上数据仅仅限于它每一笔交易的输入，但不包括它的最终状态。就比如我们从一个地方到另一个地方，我们上链的数据仅仅是怎么走的路线，而不包括我们在两个地点都做了什么。而对于 Rollup 在每次系统完成交易等行为时都会进行欺诈验证。

目前主流的 optimistic rollup 我们称之为一轮交互式 Rollup，而 Arbitrum rollup 我们称之为多轮交互式 optimistic rollup。Arbitrum rollup 则是相较于其它类型的 optimistic rollup 进行更多轮次的欺诈验证，一轮的 optimistic rollup 在解决欺诈争端时需要在链上模拟一次完整的调用即频繁的调用合约，成本较高，如果攻击者对某笔交易不计成本攻击，虽然攻击者最终会失败，但将影响到项目上其它合约的正常运行。

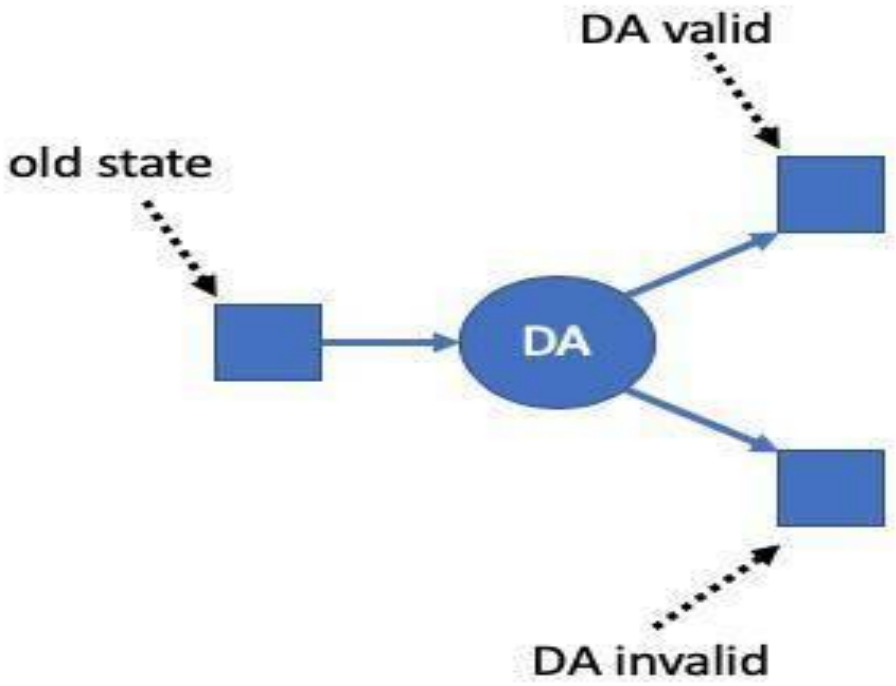
而 Arbitrum rollp 不受此限制，它会把争议范围缩小到单笔，而不会影响到其它正常合约的

执行，可以以较低成本在链上解决该争议为止，同时 Arbitrum rollp 进行多轮欺诈验证也使得在其上部署的协议更为安全

3. Arbitrum rollup 技术的工作原理

先从基础知识说起。我们使用默克尔树（Merkle Tree）来组织虚拟机的状态，因而可以算出虚拟机状态的密码学哈希值。我们把这个哈希值存储在链上，因此在协议的任何一个时间点，都会有一些虚拟机的状态（通过链上共识）被完全确认和最终敲定。这些已经获得终局性的状态的哈希值，就存在链上。

协议的参与者，通过提出一个_争议断言_（Disputable Assertion，DA）来推进该虚拟机的状态；该断言声明，从某些状态哈希开始，基于一些技术前提，虚拟机将会执行特定数量的计算步骤，生成新的状态哈希，并在执行期间完成相关的支付，生成相关的日志事件。争议断言可能是有效的（即可信的），也可能是无效的。参与者在提出争议断言时需要为保证断言的有效性而赌上一笔押金。(更多关于押注及其工作原理的内容将在后面介绍。)



争议断言会使协议产生一个决策点

如上图所示，提出一个争议断言就会产生一个协议最终必须要解决的逻辑决策点。如果争议断言是有效的，则系统将进入图中右上角的新状态，包括由争议断言生成的新状态哈希值，以及其他附带效果（产生相应的支付和日志）。若争议断言是无效的，则进入右下角的分支，争议断言被系统拒绝，原来的状态不会发生变化。

3.1 旧的 optimistic rollup 协议

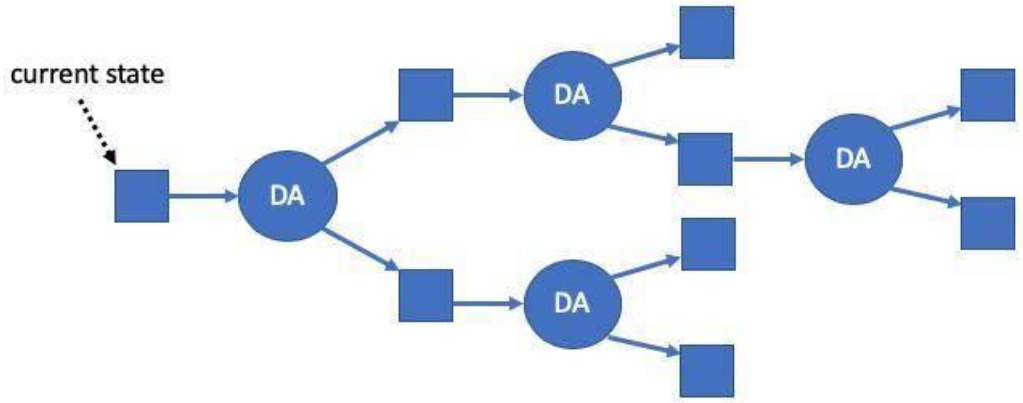
最开始的 optimistic rollup 协议每次只处理一个争议断言。当某些参与者提出一个争议断言后，会有一个挑战期，在此期间任何人都可以挑战这个争议断言。如果没有被挑战，则该争议断言将被系统接受；否则就会执行纠纷解决协议，撤销争议断言（这是为了防止提议者和挑战者合谋炮制争议结果）。

这样做很简单，但是有两个缺点。首先，因为每次只处理一个争议断言，所以虚拟机的处理速度很受限。在每个挑战期内，正常的处理流程基本上都将停滞。第二，恶意参与者通过故意挑战所有的争议断言可以彻底冻结虚拟机。攻击者需要付出一些押金作为代价，但是只要他们愿意，他们至少可以在某些特定场景下通过这种攻击长时间延误系统（而获利）。

3.2 新的 Arbitrum Rollup 改进版本

Opswap 新的 Arbitrum Rollup 协议解决了上述的两个缺点。通过 “流水线化” 处理多个争议断言，验证节点模拟虚拟机的运算速度有多快，虚拟机的处理速度就有多快。第二，我们后面将会解释，恶意参与者无法延阻系统，他们只能暂时延误对最终结果的链上确认，但这些结果对诚实节点来说早已 “无需信任地被敲定了”。所以，到底怎么做到呢？那就要讲得再深一点了……

每个状态后面最多可以接一个争议断言。如果一个状态后面没有争议断言，那么任何人都可以生成一个争议断言接在后面，作为一个新的分叉点。结果就是产生了一棵平行未来之树。



平行未来之树

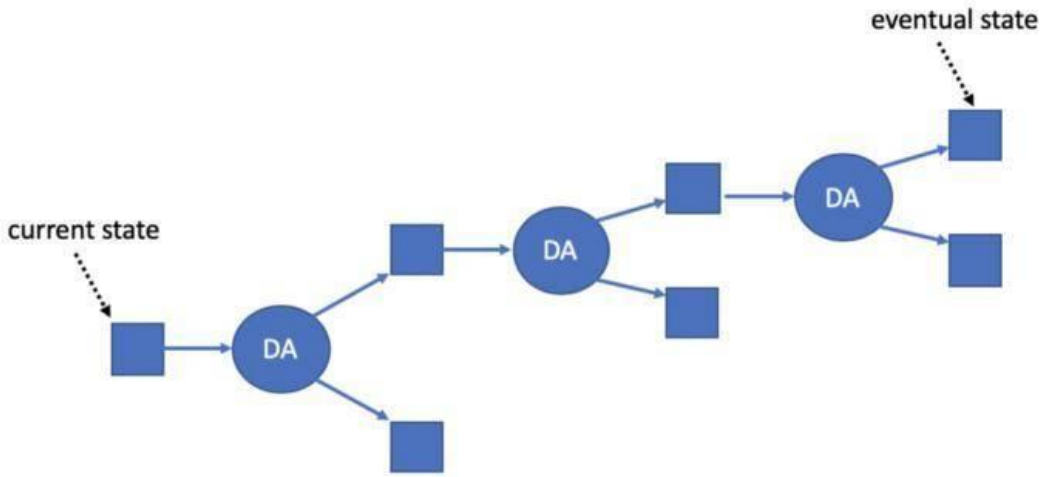
4.协议的重要部分--押注

协议的另外一个重要的部分是押注。任何人都可以在树上的某个方框（状态）后面下注。对某个方框押注，意味着你在断言该方框最终将被协议确认。换句话说，你在断言从当前状态到你押注的方框所在的这条分支是正确的分支。如果你错了，可以想见你的押金将被罚没。

押注行为不能被撤销。你可以将你的押注向右移动——可以在分叉点后向上或向下选择分支——但你不能向左移动押注，因为这相当于撤销你之前作出的押注承诺。

提出争议断言的参与者要在 “认可其断言有效” 的继任方框上押注。通常他们可以向右移动已存在的押注到满足条件的方框上。（在极少数情况下（译者注：比如他们的断言是无效的，被成功挑战）他们不能这样做，他们可以额外再押一注到需要的方框上。但是注意，他们将在相冲突的两条路径上押注，因此最终他们会损失至少一笔押金——通常自相矛盾不是明智的移动选择）。

关于押注还有一个细节：如果你押注的方框被确认成为被接受的历史的一部分，你可以选择收回押金。这意味着，如果你是正确的，你可以停止移动你的押注，直到系统 “追上” 你，然后你就可以收回你的押金。



一棵更标准的状态树 —— 由一连串的有效断言组成

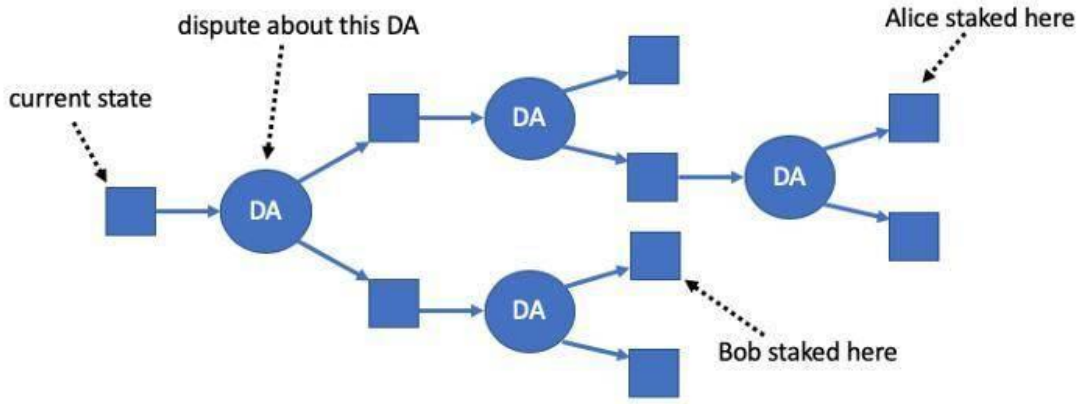
在这一点上，你可能会担心，平行未来之树会变得非常大，而且枝繁叶茂。这在现实中是不可能发生的，因为这需要多方押注不一致的结果。只要他们中有一个是诚实的，其他所有人都会损失他们的押金。更有可能的是，这棵树实际上是一个由有效 DA 串成的链，一个接一个，所有的押注都在同一条分支上。

4.1 押注期限

我们需要系统在尽可能短的时间内对每个争议断言做出决定。所以当一个争议断言被添加上链、产生一个分叉点的时候，会有一个期限与之关联。这个期限足够长，任何人如果愿意，都有足够的时间检查这个争议断言是否有效，以及产生一笔押注交易上链。任何要押注的人都必须在期限结束之前完成操作（过期的押注仍然可以上链，但它们不能决定那个争议断言的有效性）。一旦期满，所有可以决定争议断言的押注都将可知。

4.2 押注纠纷

如果 Alice 和 Bob 押注不同的方框，那么以下两件事件中，必有一件为真。要么其中一个押注可以向右移动到另一个 —— 意味着他们的断言是一致的 —— 要么找不到这样的路径。如果没有一条向右移动的路径可以连接 Alice 和 Bob 的方框，则他们必然有分歧。他们之间一定可以找到一个唯一的分叉点 —— 一个唯一的争议断言，某个人押注这个断言是有效的，而另一个押注其无效。



Alice 和 Bob 之间存在争议

当两个参与者之间出现纠纷时，系统可以启动一个交互式纠纷解决协议。我在这里没有足够的篇幅来描述这个纠纷解决协议 —— 我只想说，这是一个类似我们在其他 Arbitrum 文档中描述过的二分法交互协议。

纠纷解决协议的结果是一个参与者将被发现是错误的。这个参与者的押金会被罚没。押注会从所在的方框上删除。部分押金会给到纠纷的另一方，剩下的被烧掉。

多个纠纷可以同时解决，但是每个押注者一次最多只能参与一个纠纷。因为输家的押注将被删除，每解决一个纠纷都会减少整个系统的分歧数量。损失押金的参与者可以继续押注，但是新的押注无法影响押注期限已过的争议断言。这样做的效果是，一个争议断言的押注窗口结束后，一切有关如何处理该断言的分歧都将被消除。

4.3 押注结果确认

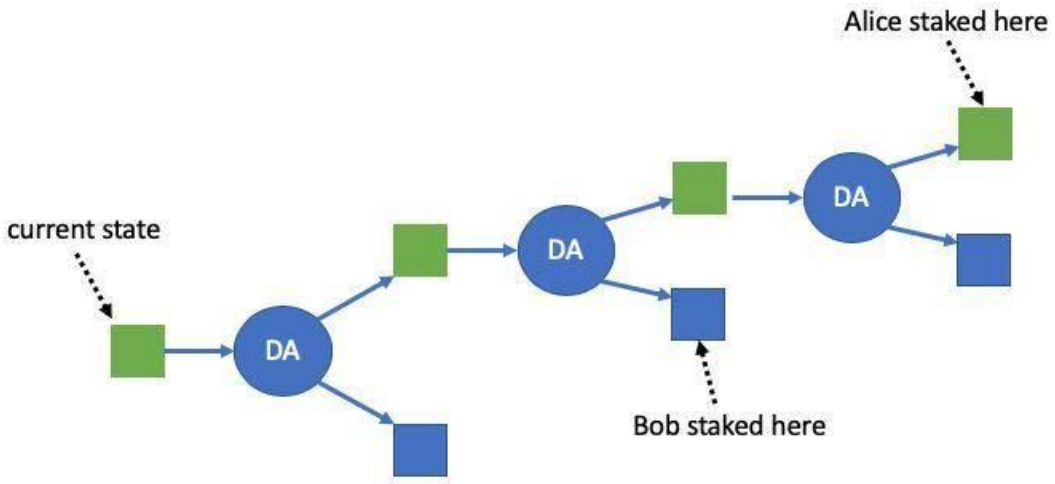
某个争议断言的押注期限到期后，如果所有及时提交（且尚未被删除）的押注，都存在于从该断言产生的同一条分支上，那么系统就可以肯定该断言的结果为真。争议断言要么被接受要么被拒绝，当前状态会移动到争议断言右边正确的方框上。如果争议断言被确认有效，则其附带效果，如支付等，也会在链上生效。虚拟机的状态就是这样向前移动的。

一般情况下，参与者都会诚实守矩，谁也不想因为押注错误而损失押金。只有有效的争议断言会被（其他参与者）断言，构成一条链，没有人会在任何争议断言的无效分支上押注。在这种情况下，所有的争议断言都会在押注期限一过后立即被确认。

5.何以无需信任（Trustless）

Arbitrum Rollup 的一个重要性质就是无需信任 —— 只要有一个诚实参与者就可以确保虚拟机状态正确向前推进。为什么呢，想象一下 Alice 总是在正确的分支上押注，如果没有争议断言，她就自己断言。

因为 Alice 总是在正确的分支上押注，所以她会赢下每一次纠纷。如果有任何人不同意 Alice，他们要么 (a) 与一个无关的第三方产生纠纷并损失押金，或者 (b) 最终和 Alice 产生纠纷并输给 Alice 押金。无论哪种情况，所有与 Alice 不一致的人都将失去押金。只有同意 Alice 的押注才能存活下去，所以 Alice 在树上的路径最终会成为唯一一个有及时押注的分支 —— 并且 Alice 的路径会被确认。



只要 Alice 是诚实的，无论别人怎么做，绿色方框都会得到确认

因为按这种方式系统是无需信任的，如果 Alice 押注某个方框，她一定知道到这个方框的路径是可信的，Alice 可以确定这个方框最终一定会被确认。对于 Alice 来说，这条路径就跟被敲定了一样。

即使你没有押注某条路径，如果你看到有好几个人对它押注，只要你相信其中至少有一个诚实的人，你就可以肯定这条路径最终一定会被确认——对你来说，这条路径就跟被敲定了一样。

5.1 无需信任的终局性（Finality）的好处

为什么说无需信任的终局性有价值？经典的例子来自于之前对其他 rollup 协议的讨论。假设一个虚拟机要向 Alice 进行支付。支付事件发生在诚实的路径上，但是包含这笔支付的方框还需要等待一些时间被链上确认。

无需信任的终局性让 Alice 可以立即拿到钱。如果 Bob 有余钱，可以立即付给 Alice，以交换 Alice 尚未被确定的收款(加上支付给 Bob 一点小费)。Bob 只有在他能确定 Alice 的

这笔收款一定会发生，才会想和 Alice 交易。Bob 可以通过押注诚实的结果来确保这一点——这样他对这笔支付最终一定会发生抱有无需信任的信心。不仅仅是 Bob 可以这样做。任何有点钱的人，都可以用这样的方式借钱给 Alice 或者有她这样需求的人，这些人通过提供更低的费用相互竞争，使 Alice 立即拿到钱的成本大大降低。

关键是，这种市场机制的可行性取决于无需信任的终局性。如果 “每个人” 都知道这件事最终会被确认，那么链上确认的延迟就不会带来那么多的不便。

不仅对于支付，虚拟机能做的其他事情也是如此。如果虚拟机要生成一个日志事件记录发生了什么事情，无需信任的终局性意味着任何人都可以肯定地采取行动，因为这个日志事件最终一定会被链上承认。

5.2 延迟攻击

因为这个系统是无需信任的，坏人无法强行制造错误的结果。他们能做的只能是延缓系统。这样做需要他们付出押金，如果押金数额巨大，则代价高昂。想象一下，如果有人宁愿付出押金也要发动延迟攻击，他们能造成的最坏的情况是怎样的？

首先要注意的是，坏人无法阻止好人继续构筑诚实的分支。而且他们也无法阻止好人获得对 “诚实的分支终将被确认” 的无需信任的信心。

攻击者能做的只是在错误的分支上押注，以延迟诚实路径的链上确认。他们每次的押注都会产生一起和诚实参与者的纠纷，而诚实的参与者会从纠纷中拿走攻击者的一大部分押金。等到攻击者的全部押金被拿走了，链上确认还会继续向前推进。

如果攻击者多笔押注错误的结果会怎样？那么这些押金会在一个接一个的纠纷中被拿走。如果有多人押注诚实的结果，这些人都可以进入纠纷解决，并行拿走攻击者的押金。而且需要注意，所有人都很清楚发生了什么，很多人都想加入进来分一杯羹，押注正确的结果从攻击者手上抢夺押金。如果诚实方有 K 个人押注，则在一次纠纷延迟期内，就要消耗攻击者 K 份押金。如果攻击者下更多的押注，那很可能会吸引更多的诚实押注者。这对攻击者来说是个灾难。

6.DEX 和 CEX 的区别

6.1 在中心化交易所，交易所控制你的资产

从你将你资产转入中心化交易所那一刻开始，这笔资产的实际控制权就已易主。在平台上账户上看到的只是交易平台呈现出来的数据，实际上的你的个人数字资产存放在交易平台的冷热钱包。

由于平台分配的地址私钥同样归你所有，你在平台账户上的币币始终都是你的，除非你要把私钥给别人。显然，这两种方式涉及到的资产安全等级是天差地别的。

在中心化交易平台,所有的交易行为均依赖于平台这个中心机构。你要提币得通过平台申请,你要交易也得通过平台完成。总而言之，中心化交易所是具有绝对话语权和操控权的。

6.2 在 DEX，智能合约解决信任问题

DEX 的交易是依靠智能合约来保证的，智能合约是一段不可篡改且无法阻止的程序，只要客观条件满足便会自动触发。“code is law”，这是程序员信奉的教条，也是区块链以技术解决信任问题的关键。

7.Opswap 的优势和交易逻辑

Opswap 彻底解决了长久以来中心化交易所存在弊端： 诸如平台跑路、限制提现,数据砸盘、插针、价格剧烈波动时掉线、卡顿等恶意的行为，最大程度地保证用户的资产安全。Opswap 为用户提供了更好的私密性、更高的资产运作透明度和监管保证的耐审查性。

与中心化交易所不同，Opswap 不控制用户的资金。相反，资金通常是由数字钱包和智能合约以去中心化的方式储存。这样，就没有一个实体充当所有加密货币的所有者，损失的风险也就低得多。

7.1 Opswap 消除中介成本，带您进入密码经济时代

由于非对称密码属于“私钥签名，公钥验证”，打破了对称密码对密码验证机构（如银行、网站等）的依赖，这也是密码共识去中心化优势的核心。

密码共识能够消解第三方信任中介，这是中本聪创建比特币的直接原因，也是是比特币白皮书的逻辑起点。

DEX 消解了中心化交易所运行的成本。当然，还有更多中心化经济的运行成本可以被消解，开启密码经济的钥匙就掌握在你我手中。

7.2 Opswap 现货交易逻辑

Opswap 使用 AMM 自动做市商技术为基础，Opswap 的重要内核采用了经典的恒定乘积(xy=k)定价公式这个公式在 Uniswap 等项目中得到了充分验证。

Swap 模式：

无需交易对手，设定适当滑点，订单会立即与市场价格直接执行交易。

本质上相当于把 CEX 的交易优势和 DEX 的交易优势相结合，产生出的燃点将会是数字货币领域的一个标志性的里程碑。

7.3 Opswap 永续合约交易逻辑

首创的基于 AMM 做市算法的现货永续合约，支持最高 10 倍的杠杆。 我们在永续合约里面添加一个保证金兑换功能。例如 用户 A 在永续合约里存入 1 个 ETH 保证金，选择 10 倍多，那么合约将利用流动性提前为其兑换 100 个 ETH 现货存放在用户的保证金里， 当 ETH 涨了，A 获利要平仓，以当时的价格卖掉 100 个 ETH，中间的差价就是 A 所获得的利润，当 ETH 跌的时候，用户 A 卖出 ETH ，那么中间将损失其相应的保证金。 全部的交易以现货兑换为基础，从根本上解决了中心化交易所数据砸盘的问题。

8.Opswap 安全性

我们始终将安全性作为 Opswap 的首要考量。由于去中心化合约的业务相对复杂，对安全性的挑战就相对更高。我们对 Opswap 现货、合约在代码安全性、保证金机制、风控等方面都做了着重的安全加强。

作为一个基于智能合约的 DeFi 产品。智能合约代码的安全性是整个产品安全性的关键。我们在开发智能合约时总是将安全性作为产品的重要组成部分，优先使用成熟稳健的技术开发我们的产品。我们对 Opswap 合约的代码做了充分的测试，对于所有可能被运行的代码分支做到了 100%的测试覆盖率。

另一方面，Opswap 与业界最优秀的安全团队合作，委托他们对 Opswap 的智能合约进行全面的安全审计。Opswap 的未来的升级和新产品都会持续委托行业内最优秀的安全团队进行代码审计。

官网：https://www.opswap.io/#/
Email：contact@opswap.io
Discord：https://discord.com/invite/rTWTZrRJKW
Twitter：https://twitter.com/opswapio
Telegram：https://t.me/opsglobalOfficial
Gitbook： https://app.gitbook.com/@opswap123/s/opswap/