



MARKETPROTOCOL

Powering safe, solvent & trustless trading of any asset

April 2018



目录

概要.....	3
MARKET 解决方案.....	4
MARKET 协议.....	7
技术规格.....	8
dApps.....	13
MKT 代币.....	14
MARKET 合约示例.....	15
团队.....	25
顾问.....	26

概要

作为一种资产类别，加密技术发展迅速，在许多应用程序中具有广阔的前景。与此同时，还伴随着很多挑战。价格波动是限制市场规模的一个重要因素，致使加密资产无法作为一种价值储存、流通手段或代币应用程序发挥效用。中心化交易所存在资金安全性、偿付能力和监管等方面的显著问题将在区块链技术的世界得到解决。

MARKET 协议（“MARKET”）创建了开放源码基础块，为以太坊区块链上的去中心化交易所和贸易活动搭建了桥梁。MARKET 是一种由参与者管理的开放源码协议，MARKET 团队不会从中收取任何费用。MARKET 平台为交易双方提供了安全、有偿付能力和免信任的市场环境下的资产买卖。

与传统的衍生品类似，MARKET 智能合约是一种介于两个或更多个体之间的合约，基于标的资产的价格而定。例如，交易者可将其数字资产作为抵押品创建 TSLA/ETH 等关联，无需将其转化为法定货币。交易之前，交易者将其担保基金存入 MARKET 智能合约中。交易过程中，此担保基金将被锁定以确保合约的可偿付能力。

第三方项目可根据协议创建去中心化应用程序（也称“dApps”）以进一步扩展其可用性。MARKET 团队准备发行首款公测版 dApp，可确保用户对 MARKET 智能合约进行部署，并搜索他人的 MARKET 智能合约。

MARKET 协议团队人员具有不同的技术与财务背景，并累计有 30 多年全球交易所电子交易经验。自 2014 年起，联合创始人 Seth Rubin、Phil Elsasser 和 Collins Brown 齐心协力共同管理 24 小时算法交易团队，促成并推动了 MARKET 协议发展成为一个开放、免信任的去中心化交易市场。

MARKET 解决方案

传统衍生品交易所

1. 保证金资金、强制清算和资金杠杆

通过 MARKET 创建的合约设立了上限和下限。例如，一旦基础价格浮动达到 50%（就如合约所设定，无论是到达上限或者下限），MARKET 将自动清算所有贸易约定，此后一个完整的合约生态环就完成了。

在 MARKET 合约交易前交易者需要提交资金。执行交易期间，每次交易的最大下限提交至抵押池中，直到贸易约定存在。

传统交易所通常采用追加保证金或强制清算进行交易活动。MARKET 可消除环境带来的系统性杠杆风险，确保交易者可以有效分配资金，通过预定义下降趋势订立贸易约定，进而防止采用危险和破坏性的市场清算活动。市场压力较大时，强制清算往往会导致产生进一步额外清算。

例如，2017 年 6 月，GDAX 上一笔数百万美元的大额沽盘导致以太坊价格从 317.81 美元骤跌至 224.48 美元，随后引发了多达 800 次的止损和保证金清算令以阻止价格进一步下跌至 0.1 美元，以期待短期内恢复市场。¹2017 年以太坊的闪电崩盘具有系统性的危害，而这种市场状况恰好可通过 MARKET 合约完全解除或阻止。

2. 清算公司和基金托管

中心化的衍生品交易所通常委托清算公司来处理客户资金。2011 年，曼氏金融报告的客户账户资金缺口高达 12 亿美元。¹曼氏金融宣布破产，约有 33,000 名客户的资金被冻结。²目前资金托管和客户资金分离主要与交易所模式有关。

3. 市场偿付能力

通常情况下，交易所可能会有意识迷惑交易对手，这就很难将整个市场视为一个整体。参与者必须相信清算公司和交易所适合监测交易者的财务状况和资本余额，但无法保证市场偿付能力。

4. 商业合约

作为营利性实体，交易所只提供他们认为可以盈利的合约。其拥有新合约的最低市场参与度和贸易约定目标以吸引更多的重要机构客户。因此，具有潜在效用的合约可能不会上市，迫使交易者和企业仅在交易所列出的合约中进行交易。

5. 首先访问/信用验证

现有交易所需要进行信用验证和限制可用性的最低账户余额。

¹ 彭博新闻 (2011)。曼氏金融短仓差额超过 12 亿美元。访问网址：

<http://www.crainsnewyork.com/article/20111121/FINANCE/111129988/mf-global-shortfall-may-be-more-than-1-2b>

² Sauer, F. (2014)。曼氏金融破产解释（为什么是一种犯罪）。访问网址：<http://www.fredsauermatrix.com/the-mf-global-collapse-explained-and-why-it-is-a-crime/>

加密货币交易所

加密货币交易所与衍生品交易所不同，前者更贴近美国股市。很多独立的加密货币交易所向参与者提供场地以进行离散交易，例如将比特币（“BTC”）兑换为美元（“USD”），或将以太坊（“ETH”）兑换为其他 ERC20 代币。

通常情况下，交易所会向用户收取交易额、押记额、提款额或法定货币存款的一定比例作为手续费，该费用取决于订单类型。由于市场参与者所在各交易所资金控制、偿付能力、货币流通性、监管制度等因素不同，因此，各交易所的加密货币价格也有所差异。

鉴于全球用户群对加密货币交易所和制度需求日益增加，芝加哥期权交易所、CME 集团和 LedgerX 开始交易比特币期权和衍生品。然而，此类项目均建立在传统交易所和清算模型之上，并且上述章节所涉及的问题仍然存在。此类项目还要求交易者缴纳美元保证金。

1. 数字借据

Coinbase 和 Kraken 等中心化加密交易所的交易不会在区块链上产生实际交易，而是交易所总帐上的内部交易。只有当用户从交易所的用户钱包中撤销加密代币时，区块链上才会产生实际交易，从而使用户可以有效控制代币。在此之前，资金处于混合交易所钱包中，使交易者的资金面临风险。

2. 有限产品

交易所存在有限产品，对于无法与比特币、莱特币或瑞波币等其他大多数加密货币进行交易的货币，上述去中心化交易所将限制其交易为 ERC20 代币。此外，所有交易均为 A 对 B 基本交易。交易者只能交易其所拥有的资产。如想披露多个加密资产的价格，则其必须购买、持有并存储各资产。

3. 无短期贸易约定

交易者对短期贸易约定加密资产具有有限期权，也可对价格下跌投注。目前，交易者没有简单的方法来买卖资产和短期交易加密资产，方法繁琐复杂。

4. 市场效率低下

通过允许投机者和套期保值者对价格下跌投注，卖空者可在市场上提供重要力量。卖空者是高效市场中的根本部分。此外，包括美国交易者在内，大多数交易者均无法部署加密资产的杠杆作用，因此，交易者必须宣布导致资本分配效率低下的未平仓交易的全部价值。

MARKET 协议

MARKET 为开发商提供了一个免信任的安全框架来创建去中心化交易所，包括必要的清算和抵押池基础设施。作为协议，MARKET 使第三方能够建立交易、买卖盘传递及相关活动的应用程序。

去中心化协议通过以太坊区块链上的智能合约来促进风险转移和免信任交易系统。MARKET 合约从标的资产、数字或现实基础资产中获得价格。不限制交易者拥有或现有 ERC20 代币，允许暴露比特币、瑞波币或门罗币等其他大多数加密货币的价格。

作为衍生品，MARKET 合约为用户提供持续的价格暴露和未来结算。交易者可以在任何流动性合约中快速订立长期或短期贸易约定。之后，贸易参与者在交易执行前向抵押池提供资金，然后，合约依照原则在规定的结算日或以当交易者在结算日之前贸易约定终止时分配资金。

清算功能性具有安全可靠的框架，以系统负责的方式管理加密资产、贸易约定和杠杆。所有智能合约和抵押池余额均可在区块链上公开获取。任何人或实体均无法控制参与者之间的资产流动、对盘、合约创建或争议解决。

参与者以民主和公平的方式管理议定书。协议的交易者应为所有者和决策者。MARKET 目的在于为用户提供最有效、最安全和最安全的环境，同时创造稳健和公平的市场。

技术规格

综述

MARKET 允许第三方通过订单托管来创建“市场”。通过收取其设置和控制的交易费用以激励订单主机（以下称为“节点”）下订单。通过智能合约，MARKET 简化了获取抵押品、验证信誉、结算执行和客户资金保管等程序的复杂性。

目前，节点不负责交易匹配和资金托管。协议交易者可以作为做市商发布订单，或作为购买者交易潜在订单。节点如同时在公告栏中发挥作用，并广播所有发送给它们的做市商订单，从而为购买者潜在匹配、最终选择和执行交易。

未来，MARKET 可能会开发替代节点解决方案，例如完全去中心化的订单簿和匹配以及其他扩展实施活动。

合约创建与清算

通过抵押池确保合约偿付能力时，MARKET 允许用户创建合约、指定条款、公布条款，并提供自动结算机制。任何交易者都可以通过规定合约规格来创建新合约。合约创建者将看到以下选项：

- **标的工具：**定价的标的资产是什么？这可能是实物资产，也可以是数字资产。
- **价格下限和上限：**用来规定参与者的最大损失或收益，以及参与者必须发布的短期或长期贸易约定。
- **合约终止日期**
- **结算机制：**创作者选择一种基于 oracle 的解决方案来决定在损益计算中所用合约的最终结算价格
- **基础代币：**定价的基础货币是什么？这决定了支付到抵押池的 ERC20 代币

合约价值来源于持有加密货币的价值，因此，任何实物资产、数字资产或 ERC20 代币均可创建合约。

共享抵押池

各 MARKET 合约均包括多个智能合约，此类合约创建了共享抵押池和规定基本代币（任何 ERC20）中记录的单个交易者余额所需要的核算。

交易者将在交易前将抵押品以 ERC20 有效代币的形式存入智能合约，所有交易利润和损失均使用代币结算。

交易者存入代币后，可以根据其智能合约余额提交订单并订立贸易约定。交易者开仓时，代币将从交易者的余额转移到抵押池。特定合约中，抵押池中的代币可全部用于弥补所有未清偿贸易约定的最大损失。如果交易者在结算前关仓，则其可提取之前分配的资本（加上或减去任何损益），或将此类资本用于进一步交易。或者，交易者可保持其贸易约定直到终止。在这种情况下

下，oracle 可提供结算价值以确定交易者的盈利或亏损。一旦确定合约，用户可归还其抵押品。

执行未平仓交易时，抵押品处于完全抵押状态，以消除交易对手风险，并取代传统交易所的核心功能之一。管理抵押池的智能合约将为传统资金托管问题提供可靠且免信任的解决方案。

执行的交易价格和数量决定了从交易者余额转移到抵押池的抵押金额。分配的抵押额等于该仓位可能出现的最大损失。对于买方，其为入场价格减去合约最小值；对于卖方，其为合约最大值减去入场价格。

```
1  /// @notice determines the amount of needed collateral for a given position (qty and price)
2  /// @param priceFloor lowest price the contract is allowed to trade before expiration
3  /// @param priceCap highest price the contract is allowed to trade before expiration
4  /// @param qtyDecimalPlaces number of decimal places in traded quantity.
5  /// @param qty signed integer corresponding to the traded quantity
6  /// @param price of the trade
7  function calculateNeededCollateral(
8      uint priceFloor,
9      uint priceCap,
10     uint qtyDecimalPlaces,
11     int qty,
12     uint price
13 ) pure internal returns (uint neededCollateral) {
14
15     uint maxLoss;
16     if(qty > 0) { // this qty is long, calculate max loss from entry price to floor
17         if(price <= priceFloor) {
18             maxLoss = 0;
19         }
20         else {
21             maxLoss = subtract(price, priceFloor);
22         }
23     } else { // this qty is short, calculate max loss from entry price to ceiling;
24         if(price >= priceCap){
25             maxLoss = 0;
26         } else {
27             maxLoss = subtract(priceCap, price);
28         }
29     }
30     neededCollateral = maxLoss * abs(qty) * qtyDecimalPlaces;
31 }
32 }
```

此抵押额始终处于抵押池中，直到交易结束。之后，合约会更新用户未平仓交易的价格和数量。

杠杆和合约范围

MARKET 合约提供标的资产的持续损益风险，直至在定义合约范围内合约创建期间指定的价格上限和价格下限。MARKET 协议提供的杠杆作用与传统杠杆作用不同，后者存在强制清算和短期贸易约定的风险。

开始新交易时，交易者将执行价格与抵押池的最大损失之间的差额。该行为所需的权益少于提供隐性杠杆的贸易约定名义总价值。合约范围内初始条目之间的所有价格均可进行交易。交易结果并非具有双重性。

如果超出范围的上限或下限，则合约与参与者一方面获得最高收益，而另一方什么都得不到（最大损失）。这两种情况都可能违反，则合约会按照惯例继续交易和终止。进而，我们会实施这一功能以确保市场仍然具有偿付能力。该过程是合约框架和 *MARKET* 协议最重要的特征之一。未平仓交易的杠杆作用支付数额取决于相对于当前合约范围的交易价格。例如，下跌较少时，用户售出价格接近最高值，则需缴纳的保证金也少。

交易者可以通过移除一系列合约来确保收益结构不达上限。我们系统通过 *MARKET* 第三方对每份合约进行多次冲击，并以一种简单经济的方式向交易者披露其想要的交易价格。由于交易者可以通过交易来订立多个合约，因此，这可能会为交易者带来套利交易机会。

两个详述的交易示例中，*MARKET* 协议文件中包含后者。

卖空

目前，短期加密资产期权有限且效率低下。然而，*MARKET* 合约可使其简单易行。

如果双方愿以预定价格交易，则可进行交易。所以，短仓没有必要寻找或借入标的资产。如果 *MARKET* 合约有效并具有偿债能力，则可卖空。

订单提交和执行

交易时，用户将首先将必要数量的基本代币提交至抵押池智能合约，从而确保可用于交易的资金。交易前，为智能合约提供资金可以减少匹配期间的失败交易数并创造更好的用户体验。

交易时，交易者会将作为做市商将订单提交给提供价格和数量的节点。一旦收到订单后，节点会确认做市商的地址在智能合约中有足够的余额来下订单。接下来，节点将在订单簿中发布和维护订单，直到另一名交易者（购买者）填写订单。提供服务时，节点设置并收取交易费用。购买者负责通过智能合约呼叫交易功能并提供他们的地址来填写做市商订单。购买者控制订单匹配，增强节点的免信任功能。该情况下，资金从交易者的智能合约余额转移到抵押池。节点从不处理资金。最后，每个参与者的新职位都记录在智能合约和区块链中。

```
1      struct Order {
2          address maker;
3          address taker;
4          address feeRecipient;
5          uint makerFee;
6          uint takerFee;
7          uint price;
8          uint expirationTimeStamp;
9          int qty;
10         bytes32 orderHash;
11     }
```

如需要多次执行，则以 LIFO（后进先出）方式终止贸易约定。某个贸易约定终止后，适当数量的抵押品（包括任何收益或损失）将被分配回用户的智能合约余额用于交易或提款。

```

1  struct UserNetPosition {
2      address userAddress;
3      Position[] positions; // all open positions (life upon exit - allows us to not reindex array!)
4      int netPosition;      // net position across all prices / executions
5  }
6
7  struct Position {
8      uint price;
9      int qty;
10 }
11

```

合约终止和结算

合约终止后，MARKET 的内置功能允许使用 Oraclize.it 或 Thomson Reuters BlockIQ 等 oracle 来进行结算。Oracles 向区块链提供外部数据，合约创建者将可设置 oracle 查询的频率。

```

1  /// @param queryID of the returning query, this should match our own internal mapping
2  /// @param result query to be processed
3  /// @param proof result proof
4  function __callback(bytes32 queryID, string result, bytes proof) public {
5      require(validQueryIDs[queryID]);
6      require(msg.sender == oraclize_cbAddress());
7      lastPriceQueryResult = result;
8      lastPrice = parseInt(result, PRICE_DECIMAL_PLACES);
9      UpdatedLastPrice(result);
10     delete validQueryIDs[queryID];
11     checkSettlement();
12     if (!isSettled) {
13         queryOracle(); // set up our next query
14     }
15 }

```

通常情况下，oracle 会每天查询是否超出了合约价格范围或者合约是否超出终止日期。如超出合约范围或超出终止日期，则合约终止，并且开始结算。另外，如果满足结算条件（例如，超出价格范围），则用户可随时调用功能来促成结算。

```

1  /// @dev checks our last query price to see if our contract should enter settlement due to it being past our
2  // expiration date or outside of our tradeable ranges.
3  function checkSettlement() private {
4      if(isSettled) // already settled.
5          return;
6
7      if(now > EXPIRATION) { // note: miners can cheat this by small increments of time (minutes, not hours)
8          isSettled = true; // time based expiration has occurred.
9      } else if(lastPrice >= PRICE_CAP || lastPrice <= PRICE_FLOOR) {
10         isSettled = true; // we have breached/touched our pricing bands
11     }
12
13     if(isSettled) {
14         settleContract(lastPrice);
15     }
16 }

```

通过 oracle 调用交易所应用程序界面，可将合约结算为任何交易活跃的 ERC20 代币、加密货币等指定资产的价格，例如，定义结算程序的合约可以在预定时间内指定 Kraken 代币的最后交易价格。

为避免错误或不准确的结算价格，我们将在初始执行（合约终止）到用户可能提取资金时间之间预留一段时间延迟。如超过一定比例，未平仓交易参与者产生结算争议，则合约进入有争议的状态。

如果产生结算争议，可以使用备用 oracle 或备用 oracle 组来获得结算价值。随着基于人群共识机制的发展，MARKET 意在实施额外的解决机制。在此之前，有争议结算也可能通过更加中心化的流程得以解决，以确保资金公平地归还给参与者，而不是永久性地陷入合约中。

dApps

为帮助非技术用户充分利用 MARKET，除第三方开发商外，MARKET 还将建立去中心化应用程序（也称“dApps”）用于 MARKET 协议。MARKET 将创建简单的用户界面，直观地解释选择合约变量和将 MARKET 合约部署到区块链的过程。此外，合约开发工程师有能力搜索以前所部署 MARKET 合约及其规格。用户还可以测试 oracle 查询，以确保其在合约部署之前按预期运行。

MKT 代币

MKT 是 MARKET 生态的基础代币，并从 MARKET 各个方面的集成中得到经济利益。MARKET 中可免费进行点对点交易，协议本身不收取任何费用。提供订单托管和管理的节点可以选择设置，并收取提供此服务的交易费用。节点不接受未缴纳交易费的订单，MKT 中的节点将收取交易费。MKT 持有者还可对协议改进和开发投票，确保使用 MARKET 协议的用户和项目对协议未来均具有发言权。

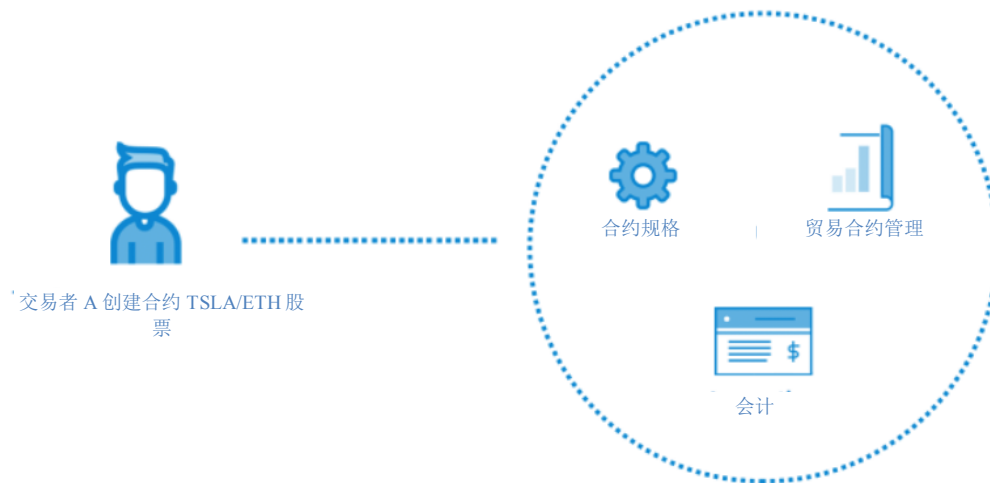
代币用户

- 1. 合约权限：**最初，参与者需要发布 25MKT 代币以及适量基本代币（用于抵押）以交易各用户自定义合约。由此发布的 MKT 在其停止交易特定合约或合约终止时返还给用户。
- 2. 交易费：**节点向 MARKET 协议的用户提供服务并可在交易所收取 MKT 中指定的交易费用。每个节点都为服务设定相应费用。预计节点将根据所提供的费用和服务进行区分，为用户提供许多执行选择。
- 3. 结算：**为了保证准确性和自主性，大多数合约均可公开访问 Oracle 解决方案以进行自动结算。如果出现结算分歧或结算过程中断，MARKET 有意采用多种解决方案，包括从 MKT 持有者池中提取的备用 oracle 或常规解决方案。
- 4. 合约创建：**最初，要求用户至少持有 500 MKT 来创建新的交易合约。此随意控股目的是鼓励创造有思想的合约。
- 5. 协议决定：**我们期望 MKT 代币持有者对协议决定和开发投票，其中包括如上所述的合约访问或创建所需的 MKT 代币数量。账户持有的 MKT 越多，则其投票影响力越大，起初，决策将更集中。MKT 持有者代表整个生态，交易者、节点和投机者，每个人都有协议方向的声音。随着时间推移，我们想要创造更加基于社区的决策模式。

MKT 合约示例

示例 A—单一股票合约

在该示例中，我们展示了如何使用 MARKET 协议创建 ETH 和传统证券之间的衍生合约，特斯拉股票（NASDAQ, TSLA）



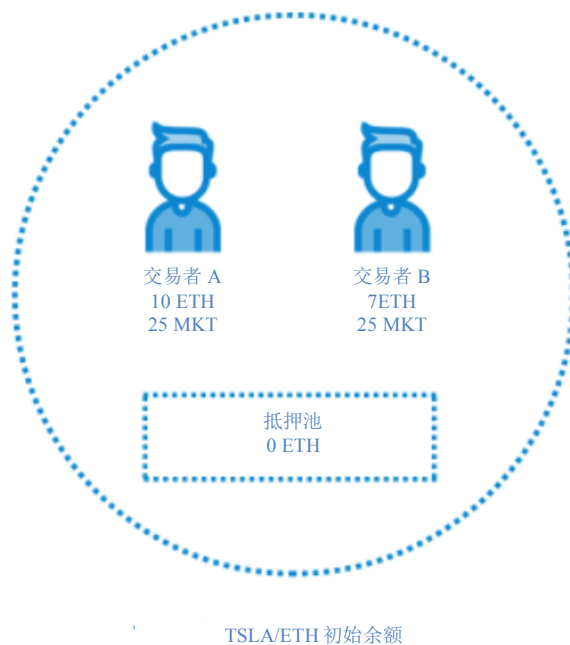
用户 A 将定义：

1. 基础货币（任何 ERC20 代币，在本示例中符合 ERC20 标准的 ETH）
2. 标的资产（在该示例中为 TSLA）
3. 结算（通过 oracle 实现）
4. 上限和下限—如果 TSLA/ETH 的价格目前为 10，则可以使用上限 15 和下限或 5 创建合约。如果合约基础工具达到 5 或 15，则到期。所有价格在 5 至 15 之间均可进行交易。

合约终止时，所有用户都有所收益或损失。

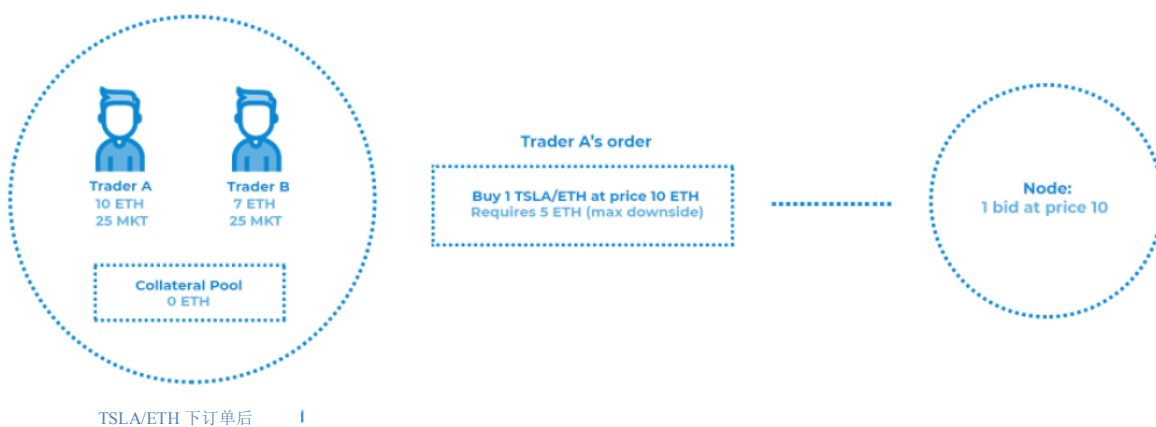
交易前

1. 交易者确定他们想要交易的合约规格，如 TSLA/ETH
2. 交易者 A 将 10 ETH 和交易者 B 的 7 ETH 存入智能合约开始交易。每人可存 25MKT。
3. 如有余额，用户可以随时提取余额。未平仓交易用的资金不符合提取条件。
4. 合约有一个共同的抵押池用于持有未平仓订单和交易的资金。



下订单

1. 交易者 A 创建一个订单对象，以价格 10 购买 1 个 TSLA/ETH，签名并将订单发送给节点。
2. 规定的 TSLA/ETH 合约中，上限为 15，下限为 5。



3. 节点确认交易者 A 具有智能合约中可用的资金来创建此订单。
4. 节点接受订单，并以 10 价格显示 1 个 TSLA / ETH 的出价。

交易执行

1. 用户 B 查看节点持有的订单，可以看到用户 A 的价格为 10 的 1 次出价订单。用户 B 想要以 10 的价格出售 1 个 ETH/Tesla
2. 然后用户 B 将用户 A 的订单调用填充交易函数到订单信息的 MARKET 智能合约
3. 之后，MARKET 智能合约填写订单并分配交易合约。
4. 更新抵押品余额和用户余额，每个用户将其最大损失添加到抵押池中
5. 为节点指定的交易费（以 MKT 代币进行）在执行时发送。



交易后

使用 TSLA/ETH 合约，上限为 15，下限为 5，有三种交易后的情况。如果 15 或 5 交易在标的 TSLA/ETH 对中，则合约终止并结算

1. 合约终止前交易者终止贸易约定
 2. 交易者持有合约终止的贸易约定
- 合约价格违反上限或下限

情形 1：交易者提前终止贸易约定

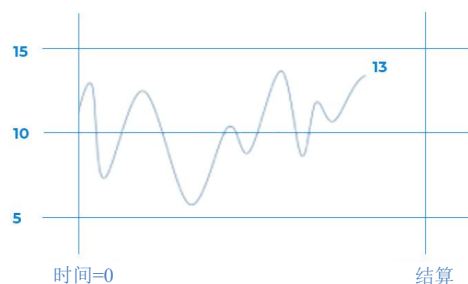
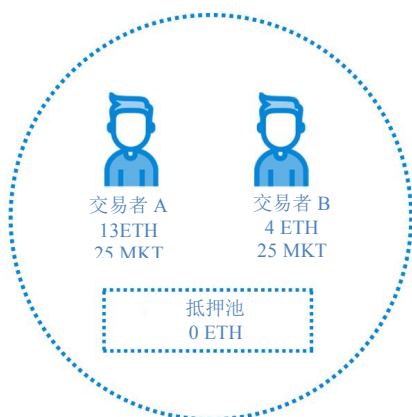
这两位交易者均进行了一项未平仓交易，A 在 10 时长期持仓 1，B 在 10 时短期持仓 1。交易价格为 10 的 TSLA/ETH 关系目前交易价格为 13。

交易者 B 想要知晓损失，在之前指示的相同过程中创建新订单，并将订单提交给节点。结账单无需额外的抵押品。

交易者 A 想要获得利润并填写用户 B 的订单。

两位交易者均为平价交易。交易者 A 已交易 3 个 ETH，而交易者 B 已经损失 3 个 ETH。

初始抵押余额均为 5 ETH。交易者 A 收回 8 个 ETH 和交易者 B 的 2 ETH。抵押池总余额为 10 ETH，现在为 0。如果合约已交易，则其均会收回 MKT 代币。

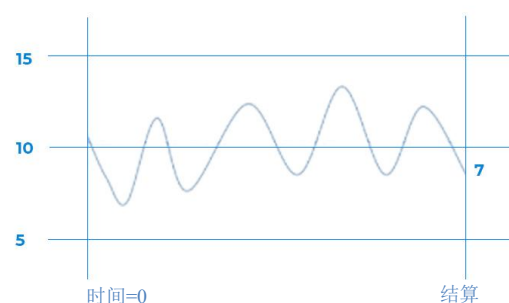
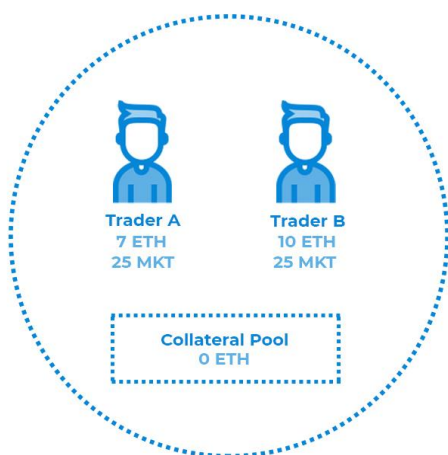


情形 2：交易者维持交易直到合约终止，从合约创建起 1 个月

这两位用户均进行了一项未平仓交易，A 在 10 时长期贸易约定 1，B 在 10 时短期贸易约定 1。合约进行有机结算。Oracle 根据用于确定交易者的 PNL 的合约规格交付必要的结算价格。

在这种情况下，依照 7 进行合约结算。

用户 A 已失去 3ETH，并收回 2ETH（初始存款+PNL），并且用户 B 有 3ETH，收回 8ETH（初始存款+PNL）。

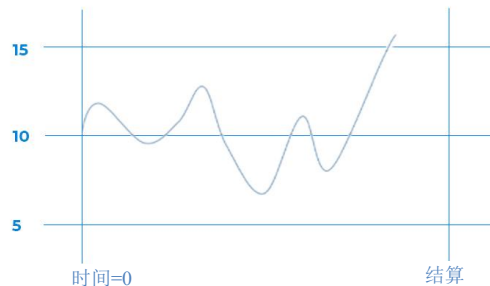
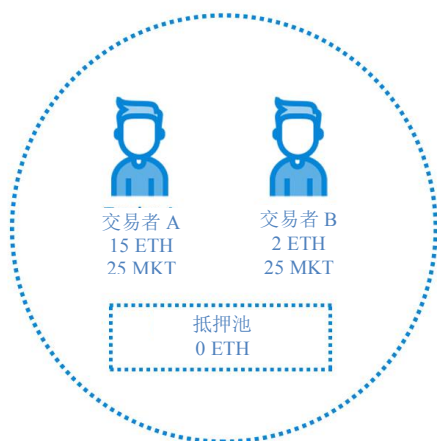


情形 3：合约达到上限 15

这两位交易者均进行了一项未平仓交易，A 在 10 时长期持仓 1，B 在 10 时短期持仓 1。

在这种情况下，合约达到交易上限 15，当这种情况发生时，合约自动以 15 的价格终止。

交易者 A 有 5ETH，并收回 10ETH（初始存款+/-PNL），并且交易者 B 失去 5ETH，收回 0ETH（初始存款+/-PNL）。



所有未平仓交易均记录在区块链中，且对所有用户都是透明的。抵押池余额应全部用于支付所有未平仓交易。当用户进行未平仓交易时，会计采用先进先出的方式完成。

示例 B—有效代币对冲

用户可以用 MARKET 对冲有效代币，从而消除价格上涨和下跌所带来的影响。现有和未来的多数 ICO 代币均会为所有者提供一些好处或效用。然而，此类代币可能会有相当大的价格波动，并可能会超过与代币相关的任何潜在利益。MARKET 为有效代币的所有者提供了一种方法来抵消其价格风险，同时保持与拥有代币相关的效用。代币所有者从不出售，且转让其代币。

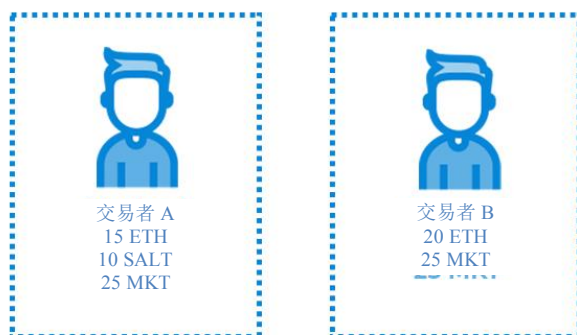
该示例中，我们将使用 SALT 借贷代币。SALT 是一个端对端借贷平台，允许用户通过加密支持的贷款借入法币。SALT 贷款代币是参与平台和获得贷款的必要因素。

自从发行以来，SALT 代币已经从 2 美元的低估升高至 17 美元。通过 MARKET，交易者可以对冲此类价格波动。

如需解释该示例，假设交易者 A 拥有一定数量的 SALT 代币并想要对其价格风险进行套期保值：

合约规格 SALT/ETH

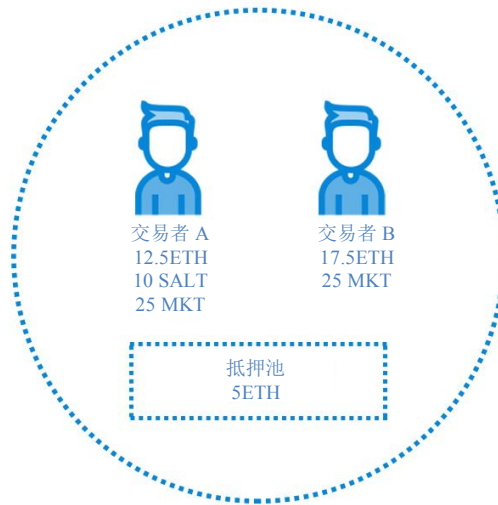
- 1 份合约=1SALT
- 上限=0.75ETH
- 下限=0.25ETH
- SALT/ETH 以 0.5 ETH 交易，表明每个 SALT 代币价值 0.5 ETH



交易前

1. 交易者 A 拥有 10 个价值 5 ETH 的 SALT 代币，如需对冲，则需要售出 10 份合约。
2. 交易者 B 未持有 SALT 代币，但想要推测 SALT 代币的价格

应注意，交易者 A 仍持有其 SALT 代币，并可以使用此代币进入平台。交易者 B 未持有过 SALT 代币，目前依然未持有。



交易者 A 他需要售出 10 份合约来对冲他的 SALT 代币价格风险。当前价格是 0.5 ETH。这表明，最大下跌可达到 $0.25 \text{ ETH} \times 10 = 2.5 \text{ ETH}$ 。

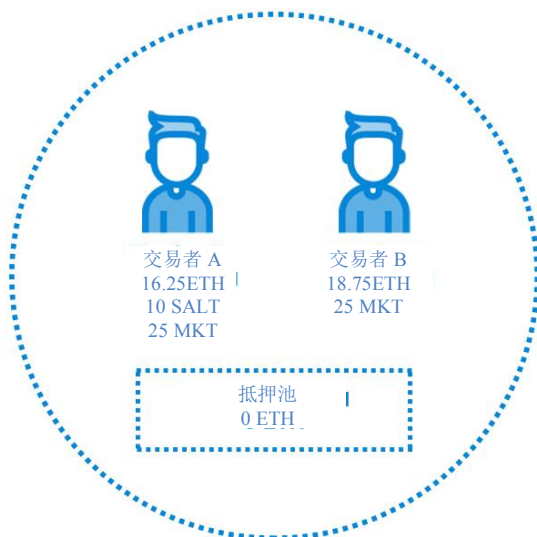
交易者 B 愿意以相同的价格购买 SALT/ETH。该情况下，SALT 代币可以上涨或下跌。

交易后

指明交易后的两个情况：

1. SALT 减值。
2. SALT 增值。

情形 1：SALT 从 0.5 下跌至 0.375，则交易者终止贸易约定。



交易者 A 获得 0.125×10 合约 = 1.25 ETH

交易者 B 损失为 0。 125×10 合约 = -1.25 ETH

交易者账户余额已更新。

交易者 A 初始 SALT 代币价值 5 ETH

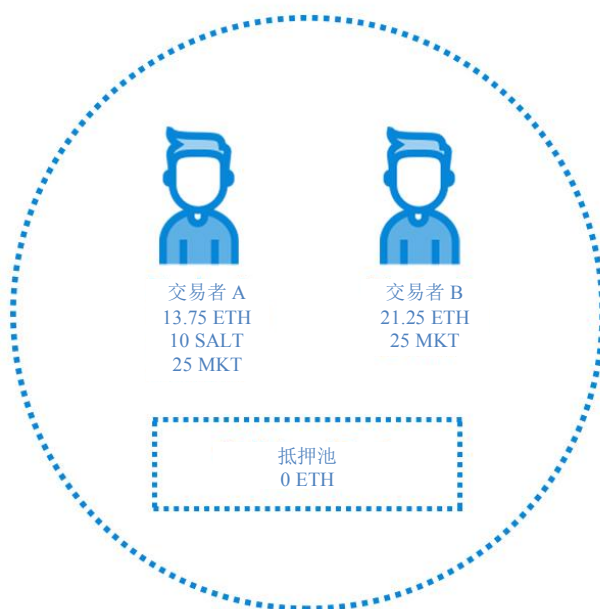
(0.5×10)，此后价值下跌至 3.75 ETH (0.375×10)。

交易者 A 在 SALT/ETH 合约获得 1.25，但实际代币持仓损失 1.25。

如果交易者 A 未通过 MARKET 对冲其持股，则其将面临亏损。

交易者 B 从未拥有任何 SALT 代币，但通过 SALT/ETH MARKET 合约购买了相当于 10 代币的等价物，则该等价物从 0.5 ETH 降至 0.375 ETH，并产生 1.25 ETH 的损失。即使未采用初始代币，该等价物与其拥有 10 个 SALT 代币相同。

情形 2：SALT 从 0.5 上涨至 0.625，则交易者终止贸易约定。



交易者 A 损失 0.125×10 合约 = 1.25 ETH

交易者 B 获得 0.125×10 合约 = 1.25 ETH

交易者账户余额已更新。

交易者 A 初始 SALT 代币价值 5 ETH

(0.5×10)，此后价值上升至 6.25 ETH (0.625×10)。

交易者 A 在 SALT/ETH 合约损失 1.25，但实际代币持仓获得 1.25。

交易者 B 从未持有任何 SALT 代币，但通过 SALT/ETH MARKET 合约购买了等价物与 10 个代币价值相同，此等价物从 0.5 ETH 增加到 0.625 ETH，并产生 1.25 ETH 的收益，即使未采用初始代币，该等价物与其拥有 10 个 SALT 代币价值相同。

团队

MARKET 协议团队人员具有不同的技术与财务背景，累计有 30 多年全球交易所电子交易经验。自 2014 年以来，联合创始人 Seth Rubin、Phil Elsasser 和 Collins Brown 始终团结合作，共同管理 24 小时算法交易团队，并在 2015 年开始从事加密货币交易，这使团队能够预测到区块链如何解决传统交易所模型和加密交易所模型固有的许多问题。此经验促成并推动了 MARKET 协议的发展成为一个开放、免信任的去中心化贸易市场。



Seth Rubin

联合创始人兼首席执行官

seth@marketprotocol.io

[领英](#)

Seth 主要负责将实用策略纳入 MARKET 协议执行和开发。自 2005 年开始担任衍生品交易者，他已经开发并管理了多个算法交易平台，作为注册做市商开展业务，他还参与了众多产品的发布会。Seth 于 2015 年首次作为交易者进入加密领域，后期专注于加密技术。他还与 Collins 和 Phil 研发并成功实施了套利交易和相关价值加密策略。Seth 对中心化和去中心化交易以及交易所结构理解深刻到位。



Collins Brown

联合创始人

collins@marketprotocol.io

[领英](#)

Collins 在 Transmarket Group 开始他的职业生涯，已在全球交易所从事各种资产类交易近 13 年，并在 2014 年，联合创建了 BRE Trading。作为职业交易者，他了解交易所用户的需求。Collins 认为，MARKET 同时具有中心化交易和去中心化交易所的优势和弱点非常重要。

他的开发团队成功解决了各类复杂问题。Collins 指导加密市场微观结构贸易和研究项目，拥有质保和生产方面的算法和软件测试经验。



Phil Elsasser

联合创始人，首席技术官

phil@marketprotocol.io

[领英](#)

过去 7 年，Phil 一直担任算法交易平台的首席开发人员。Phil 领导开发团队专注于交易基础设施、用户界面、执行平台和定量交易分析的创建。Phil 拥有实施技术解决方案所需的技能、创造力和激情，能够解决去中心化市场带来的挑战。

他还为 CME、ICE、TOCOM、SGX 等其他期货交易所编写了低延迟连接和交易策略。通过在衍生品领域的经验，Phil 还为许多加密交易所编写了策略。

目前，Phil 是加密资产对冲基金的技术和安全顾问。



Lazar Jovanovic

营销/品牌大使

lazar@marketprotocol.io

[领英](#)

作为加密资产交易者和爱好者，Lazar 已经评估了许多新项目，并完善更成功的加密技术。他还积极参与了一些创业公司的区块链发展战略和社区支持。



Mauzy Keshavarzi

社区经理

maz@marketprotocol.io

Maz 在开成功发和实施在线社区推广项目中具有丰富的经验，曾担任过多家创业公司的商业顾问。他加入了 MARKET 协议团队，热心分享加密交易和资产管理方面的经验，帮助他人。Maz 希望帮助社区了解 MARKET 的全部内容。



Perfect Makanju

软件开发工程师

perfect@marketprotocol.io

[领英](#)

为了让人们生活更美好，Perfect 对 Android、人工智能和智能合约相关的智能解决方案构建充满兴趣，他还具有丰富机器学习、图像处理和 Web 应用程序（使用 php 和 nodejs 的后端）经验



Eswara Sai

软件开发工程师

eswara@marketprotocol.io

[领英](#)

Eswara Sai 是一位热情的前端开发人员，精通 HTML、CSS 和 JavaScript。他经常使用 ReactJS 和 AngularJS 等 JavaScript 框架，用于构建动态单页面 Web 应用程序。通过与 Phil Elsasser 和其他团队密切合作，为 MARKET 研发了一款 Dapp，用于部署和测试 MARKET 合约。



Przemyslaw Szulczynski

软件开发工程师

przemyslaw@marketprotocol.io

[领英](#)

当 Przemyslaw 9 岁时，他的父亲为他买了 Commodore 64，并带有 5.25 英寸软驱。当时关于这个问题的书籍并不多。唯一一本是 Commodore 64 Memory Map，他已经阅读很多遍。在他获得了他的第一台 90Mhz 个人电脑后，Przemyslaw 在互联网花了两年时间，而却没有进行互联网连接。他正在浏览网站的离线镜像副本，有时会使用超链接。

经过多年以及开发了多个项目后，他仍然是一个技术爱好者。在跨国环境中与优秀团队合作。对数字电视、MPEG2 / H 有深入的了解。264、HDCP、嵌入式设备、导航设备、高级路由算法以及监管管理程序等自动项目。他还翻译了一本书，撰写了多篇关于 MQL5 的文章。对区块链和衍生品交易感兴趣。相信 MARKET 协议将有一个光明的未来。

顾问



Patrick Charles

数据科学和分析管道架构师

[领英](#)

Patrick Charles 拥有超过二十年的软件开发经验，涉及金融、教育、医疗保健和计算机安全等多种行业。他曾担任过技术领导者、顾问、软件架构师、工程师和研究员。Patrick 是一位开源贡献者，撰写了大量技术论文，是一位拥有两项美国专利的发明家，并且是即将出版的书籍《网络学习前沿》开篇的合着者。



Josh Fraser

Origin Protocol 联合创始人

[领英](#)

Josh 在 10 岁时开始编码。创办 Origin 之前，他共同创办了另外三家风险投资公司：EventVue、Torbit（被沃尔玛实验室收购）和 Forage。



Dan Horowitz
1010data 工程高级副总裁
[领英](#)

Dan 担任工程副总裁，监督 1010data 系统的开发。致力于使 1010data 成为网络上最快、最可靠的大数据发现和共享平台，并专注于规模化解决方案的开发工作，以世界一流的可靠性和快速查询响应时间来管理拍字节数据。在 1010data 任职 8 年期间，Dan 曾在公司的许多关键开发项目中工作，包括万亿行电子表格和各种 MBS 数据产品。在加入 1010data 之前，Dan 是埃森哲全球架构团队开发人员，负责构建定制企业管理软件。Dan 拥有罗切斯特大学计算机科学学士学位。



Casper Johansen
Spartan 集团联合创始人
[领英](#)

Casper 是 Spartan Group 的联合创始人，该公司是一家领先的区块链咨询和投资公司，将加密世界和传统金融行业的深层联系结合起来，为利用区块链技术推动跨行业中断的公司提供解决方案。



Brian Shields
Coder 公司联合创始人
[领英](#)

Brian 在债务和资本市场、数字营销和技术创业方面拥有十多年的经验。作为公认的创新者，他拥有优秀的商业头脑，与财富 500 强和 Inc.500 强公司以及早期创业公司有多年合作经验。Brian 为 Coder 的风险投资、领导销售、营销、财务和战略等方面担任顾问。Coder 成立之前，Brian 担任 Brafton 的副总监，Brafton 是一家数字营销机构，在波士顿、芝加哥和旧金山均设有办事处。Brian 毕业于伊利诺伊大学香槟分校，并获得金融学学士学位。



Brent Traidman
Bread 首席营收官
[领英](#)

Brent 拥有 15 多年的高影响力及快速增长软件公司的领导经验，其中许多已经成功退出。目前，Brent 是 Bread（BRD）的首席营收官，BRD 是全球增长最快的加密金融平台之一。Bread 被认为是密码领域的思想领袖，在 140 多个国家拥有超过 100 万的用户。在加入 Bread 之前，Brent 曾在 Vista Equity Partners 投资组合中工作过，他帮助推动两个出口。Brent 建立收入引擎并帮助制定增长战略。在不工作时，他会帮助硅谷风险投资公司就早期投资提供建议，并定期发表讲话，导师和参加全球加速器会议。