# 达摩：一种通证化债务发行的通用协议

## 摘要

达摩是一种允许去中心化发起、承销、发行、管理通证化债务资产的高度通用化、非约束性结构的协议。该协议旨在构建一种通用信息接口，通过该接口，交易所、经纪人和交易员能够合理地对通证化债务违约风险进行定价，而不需要依赖一个单一的、中心化的数据中介。达摩债务发行计划利用两类公共角色（承销商与转包商），他们在不同的市场竞争补偿费用。承包商是债务违约风险的可信发起人和评估人，转包商用无需信任的方式来促进债务的融资和发行。可以基于历史资产绩效对承销商和转包商进行经验性地评估，同样地，市场有清晰的信号来评估由任何指定的承销商或转包商证实的通证化债务违约风险。达摩债务发行流程只需要执行一次区块链上的交易，达摩深受0x协议机制的启发。

## 介绍

声明：区块链存在一个未被充分认可的优势，它们必会创建通用且无访问限制的标准以用于通证化资产分类。

截至本文撰写时，单单在2017年token sale(即ICO，Initial Coin Offering，现更多称呼为crowd sale或donation，可理解首轮首币或通证预售，与公司首轮募股类似用于筹集启动资金)就已经众筹超过20亿美金。这能证明的是对于市场对众筹产品的需求不足，然而这些产品可以补偿拥有相似风险资产[1]的散户投资者。然而传统的股权众筹机制明显早于ICO现象出现，如何解释突如其来这么多人涌入？如果通证销售在技术上可行并且在生产中已经存在多年，那么ICO生态系统在以太坊技术方面的独特优势是什么呢？ 我认为答案很简单：ERC20标准创建了共同的准则，在这个准则上，以开放和可相互操作的方式构建用于通证的二级市场的多样化生态系统。单从投资者流动性来看，通证众筹是对股权融资现状的阶梯式改进。

然而在现存的金融体系中，与债务融资领域的领头对比，股权融资筹资的资本总额显得微不足道。但是债务市场仍然保持不透明和所有权归属，无论是公开募股还是私人投资者，执行债务融资都与执行股权融资一样具有定制性和低效率。那么为什么不考虑将通证销售模型用于债务筹款？

举一个例子，假设公司发行债券作为”ICO债务“并代表所有权与ERC20通证在一个通证众筹中出售。从表面上看，以开放通用通证作为标准的债务资产所代表的世界也会在流动性和透明度方面改善现状。 然而，为了产生类似的免许可方式流动的二级市场，投资者需要一种标准化的机制来定价债务资产。虽然类似的通证是通过协议，项目或实体的挂钩价值，但类似债务的通证价值通常与匿名交易方的历史财务活动挂钩。 因此，ERC20标准不能捕捉债务资产的强制性语义，因为它没有提供以下方式：

1.检 索返回机器可读的，与资产相关的债务特定元数据（例如本金，利率）；

2.以债务资产的条约检索债务人和债权人之间的付款历史；

3.将违约风险一同定价包含于债务资产的价值中；

**达摩协议旨在弥合这一差距，并提供一种无需许可的通用机制，通过该机制，可以发行，出售，管理和定价灵活类型的债务资产，而无需依赖任何类型的数据经纪人。**

## 架构

达摩协议通过智能合约，监护员市场和标准化界面定义了使用一系列发行，资助，管理和交易债务资产的程序。达摩在0x分散式交换协[2]的设计中受到了极大的启发，使用0x广播订单消息机制作为我们达摩债务订单的蓝图，即达摩协议中的类似等价物。 该机制将在本文中进一步明确形式化。 该协议旨在支持EVM区块链，但可以扩展为支持所有具有通用智能合约功能的区块链。 首先，我们定义了一些术语。

### 代理 Agents：

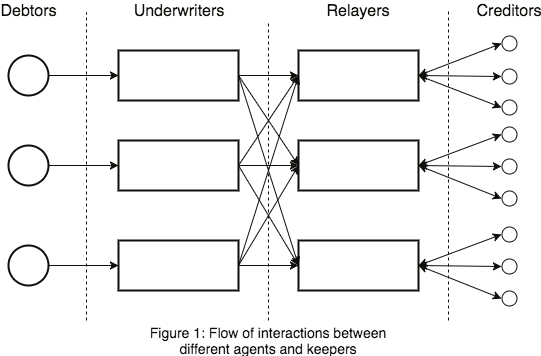
我们将代理定义为协议的最终消费者 - 即希望借入或借出加密资产的实体。 这些实体，无论是人员，公司，合同还是自动机，通常可以分为以下两类：

1.债务人 - 债务交易中借入资产并欠下债权人约定价值的一方。

2.债权人 - 债务交易中借出资产，并让债务人承担约定的价值的一方。

### 监管员市场 Keeper Marketplaces

我们采用了一个包罗万象的术语“监管员”[3]，来涵盖那些为网络提供增值服务并在各自的市场上竞争以获取手续费的服务提供商或价值用户。



### 承销商 Underwriters[4]

在传统的债务市场中，承销商是收取管理债务公开发行和将借款人违约风险定价到资产中的费用的实体。在达摩协议中，这一定义被扩展和正规化。承销商是收取由市场决定的费用的实体，他履行如下功能：

* 发起来自借款人的债务订单；
* 与潜在的债务人谈判、确定债务条款（如期限、利息、摊销）；
* 以加密方式提交他们认为该债务最终违约的可能性（Specification中详细描述了该流程）；
* 通过将债务转发给任意数量的转包商来管理债务的资金；
* 偿还债务 - 即在承销商的合理权力范围内尽一切努力确保按照商定的条款及时还款；
* 在违约或拖欠的情况下，通过法律机制收集抵押品（如果有债务）或个人资产，并将收集的收益转移给投资者；

这与大多数在线贷款方在日常承销和偿还业务中所做的事情没有什么大不同。我们认为达摩协议将为那些有抱负的在线借贷平台提供另一种更低价的途径，来促进他们的业务，并获得与现状类似的利润。他们将成为承销商，与此同时不再需要担心资产负债表风险同时也避免了开展业务之前，从传统投资者那里筹集必要的债务工具，所需要的时间和资本成本。

例：Alice对于如何为一些有抱负的ZCash矿工提供发起、承销和偿还贷款有一些新的想法，这些旷工往往需要大量的前期资金来购买批量GPUs。除了去找传统固定收益投资者，而是决定成为一名达摩协议的承销商。她获得了必要的贷款从业执照，为矿工建立了提供借贷广告服务的网站，并在ZCash社区为她的信用产品大肆宣传。当借款人来到她的网站时，他们的信用会被Alice的商用技术自动评分，他们会根据Alice的要求给出贷款条款。在接受了条款之后，Alice会以加密方式证明借款人违约的可能性，将已签署的债务订单转发给了一个转包商（relayer），并基于贷款总额上抽成了她想要的费用。整个资金的流动是透明的，可审计的，而Alice在偿还和收取债务方面的能力权限也可以根据其过往的历史记录来确定。

### 转包商 Relayers

达摩协议中的转包商与0x协议中的转包商执行类似的功能，即转包商汇总签署后的债务订单信息，按照约定的费用将信息托管在统一的账本中，并通过填写已签署的债务订单来提供散户投资者投资其要求的债务订单的能力。与0x协议中转包机制类似，达摩协议中转包商不需要持有任何代理的通证，他们只是为债权人提供一种机制来浏览汇总的已签署债务订单消息，债权人可以使用该机制无需信任的发行他们自己的债务通证，用以交换客户端侧合约交互所需要的资本（此机制将在本文后面详细说明）。达摩协议和0x协议中转包商的主要区别是：

1.达摩协议转包商没有托管一个二级市场账本，而是托管一个包含尚未发行的债务请求的账本；

2.达摩协议转包商给债权人提供已签署的特定债务的与债务订单信息及其承销商相关的元数据，以便他们能够就特定债务订单的风险状况作出明智的投资决策；

3.达摩协议中的转包商不允许任何匿名方发布已签署的债务订单到他们的账本中，他们使用裁量权，只接受来自已知的、可信的承销商的已签署债务订单。

例：Bob希望建立一个零售贷款投资者的门户，用户可以通过该门户投资各种债务资产。 如果愿意，可以使用Kayak进行点对点贷款。Bob通过建立一个在线订单簿，建立零售投资平台，允许投资者浏览债务，请求并检查债务人的信誉和支持承销商身份相关的数据，成为达摩协议转包商。Bob基于看到的Alice之前证明资产的历史表现与她的预测一致，并且知道Alice的公司是一个受公众信任和受监管的实体，Bob允许Alice在他的订单薄上广播已签名的债务订单。当一个债务订单在他的平台上成交时，Bob就收到签署债务订单所规定的手续费。

## 智能合约

达摩协议利用了几个部署在以太坊网络上的智能合约，这里只强调其中几个与理解协议机制特别相关的合约。

### 1. Debt Kernel 债务核心合约

债务核心是一个简单的智能合约，它管理所有与铸造非同质化（non-fungible）债务代币相关的业务逻辑，维护债务代币与其相关的债务条款合约之间的映射，处理将还款从债务人转移到债权人，和相关手续费发送给承销商和转包商。这些机制在介绍债务生命周期的上下文中更容易定义，在下面的规范中将得到广泛的阐述。

### Terms Contract 条款合约

条款合约是以太坊上的智能合约，是债务人和债权人就共同的、能正确就偿还条款达成一致的手段。通过拓展，合约将揭示出一种标准的接口方式来登记债务人的还款，并可以通过编程程序在贷款期限内和之后查询贷款的偿还情况。单一条款合约可以被重复用于任何遵守其还款条件的债务协议——例如，一个定义了简单的复合利息偿还计划的条款合约可以由任意数量的债务人和债权人承担。下面的规范中将具体定义该接口。

注：承诺贷款条件的另一种方案是承诺在链上提供的文本形式的贷款条款（即”Ricardian合约“[5]），然后在链下对债务人的申请进行评估。出于某些原因，我们故意不推行这种方案。首先，明确定义债务条款的通用模式会在协议中限制债务资产类型的范围，而通用的协议接口则为无限的债务条款合约打开了大门。此外，在链上承诺条款合约将消除对贷款还款状态的评估的任何不确定性——合约是一个单一的、程序化的、不可变的事实来源，它是由合约和客户端查询的。最后，拥有一个链上的偿还状态数据提供源，将大大简化违约情况发生时进行收集和偿还抵押债务的机制。

### 3. Repayment Router 还款路由合约

还款路由合约的建立是为了无需信任地将还款从债务人转移到债权人(即债务通证的所有者)。另外，还款路由合约还可作为一个可信的Oracle，向与给定债务协议相关的条款合约报告每一笔还款发生时的具体细节。这使条款合约可作为一个可信赖的接口来确定债务的违约状态。

## 规范 (Specification)

### 概述

当向债务核心合约提交一份已签名的债务订单信息时，整个债务发行过程将同步地发生在一个链上交易中。如果消息符合下面的规范定义，那么该交易会有以下过程：

* 债务人对所选择的条款的遵守和承销商对违约可能性的预测都将被提交到链上；
* 一枚非同质化、不可分割的债务通证，将被发行给债权人同时映射到上述承诺；
* 本金从债权人转移到债务人（扣除费用后），任何监管员的费用也同样的从债权人处扣除。

过程如下所示。首先，我们在协议中定义了数据包的格式。

### 消息类型 Message Types

协议中不同的代理和监管员之间的通信，是由我们称为债务订单（Debt Orders）的数据包组成的。

### Debt Orders 债务订单

债务订单是由转包商点列出的数据包，它们是达摩协议的基本元素：向债务内核提交有效的债务订单，可以触发债务通证的发行，并与所求的本金金额进行交换。达摩协议不关心债权人、债务人、承销商和转包商之间交流和转让债务订单的方式，债务订单最多可以有3个ECDSA签名：债务人的签名、债权人的签字和承销商的签名。

他们签名的具体数据，取决于他们在整个交易中所扮演的角色：债务人和债权人被要求签署债务订单的哈希(即债务订单哈希)，而承销商只需要在债务订单中签署部分数据，我们称之为承销商承诺（underwriter commitment）。

此外，并非所有3个签名都必须在债务订单上才能提交给债务核心：如果一个代理或监护员没有参与交易(即在债务订单中他们的地址为空)，或者他们参与了，但同时也是提交债务订单到债务核心合同的一方，那么他们的签名就不是必须的。

债务订单由下列字段组成:

| **字段** | **数据类型** | **描述** |
| --- | --- | --- |
| kernelVersion  （内核版本） | address | 债务核心合同的地址。协议升级时，这个地址将被更新。 |
| issuanceVersion  （发行版本） | address | 与此发行承诺关联的还款路由器合同的地址。 |
| principalAmount  （本金数量） | uint256 | 债务人请求的本金总数量。 |
| principalToken  （本金通证地址） | address | 本金通证的ERC20合约地址。 |
| Debtor  （债务人） | address | 请求借贷的债务人地址 |
| debtorFee  （债务人手续费） | uint256 | 将从债务人的本金中扣除的总手续费。请注意，债务人和债权人支付的费用总额必须等于支付给承销商和转包商的费用总额。 |
| creditorFee  （债权人手续费） | uint256 | 债权人在本金基础上需支付的额外手续费。请注意，债务人和债权人支付的总手续费应等于付给承销商和转包商的全部费用。 |
| Relayer  （转包商） | address | 列出了给定债务但得转包商地址 |
| relayerFee  （转包商手续费） | uint256 | 当债务人与债权人的关系最终确定时，债务核心将支付转包商的手续费。 |
| Underwriter  （承销商） | address | 希望为该债务资产评级的承销商的地址。 |
| underwriterFee  （承销商手续费） | uint256 | 当债务人与债权人的关系最终确定时，债务核心将支付承销商的手续费。 |
| Underwriter  RiskRating  (承销商风险评级） | uint32 | 承销商给出的债务人期望对单位价值债务偿还得平均可能性评估。必须介于0和1之间，编码为一个无符号整数，表示有9位小数（也就是说，50%的可能性将被表示为500000000） |
| termsContract  （条款合约） | address | 定义了债务还款的条款细则得条款合约地址。 |
| termsContract  Parameters  （条款合约参数） | bytes32 | 条款合同中与偿还条款相关的具体参数（如本金、利率等）。 |
| Expiration  Timestamp  （过期时间） | uint256 | 超过这个时间点交易或订单还未成交则失效。 |
| Salt  （混淆参数） | uint | 伪随机混淆参数，用于区分有相同参数的不同订单的哈希值 |

### 债务发行承诺 Debt Issuance Commitments

债务发行承诺是我们单独考虑的债务订单数据包的子集，以便为任何给定的债务协议定义规范的唯一标识符。债务发行承诺表达了债务人(和承销商)想要发行一种非同质性的债务通证，其中债务通证将不可修改地与元组(TC, P)相关联，TC是一个已部署的条款合约的地址，该合约遵守了条款合约接口(见下文)，P代表了在TC合约中所使用的一组参数。另外，承销商承诺一个值R，代表承销商对债务人将按照(TC, P)所定义的任何给定的单位价值进行偿付的平均可能性的评估。该子集数据包的哈希被称为**发行哈希**（issuance hash），在整个协议中被用作债务协议的规范唯一标识符。

| **字段** | **数据类型** | **描述** |
| --- | --- | --- |
| issuanceVersion  （发行版本） | address | 与此发行承诺关联的还款路由器合同的地址。 |
| Debtor  （债务人） | address | 请求借贷的债务人地址。 |
| Underwriter  （承销商） | address | 希望为该债务资产评级的承销商的地址。 |
| Underwriter  RiskRating  （承销商风险评级） | uint32 | 承销商给出的债务人期望对单位价值债务偿还得平均可能性评估。必须介于0和1之间，编码为一个无符号整数，表示有9位小数（也就是说，50%的可能性将被表示为500000000）。 |
| termsContract  （条款合约） | address | 定义了债务还款的条款细则得条款合约地址。 |
| termsContract  Parameters  （条款合约参数） | string | 条款合同中与偿还条款相关的具体参数（如本金、利率等）。 |
| Salt  （混淆参数） | uint | 伪随机混淆参数，用于区分有相同参数的不同订单的哈希值 |

### 债务人/债权人承诺哈希 Debtor/Creditor Commitment Hash

债务人/债权人承诺哈希是有债务人或债权人签名的有效数据，以表明他同意该债务订单上的参数。它由以下债务订单参数子集的Keccak256哈希组成:

| **字段** | **数据类型** | **描述** |
| --- | --- | --- |
| kernelVersion  （核心版本） | address | 债务核心合同的地址。协议升级时，这个地址将被更新。 |
| issuanceHash  （发行哈希） | bytes32 | 与此发行承诺关联的还款路由器合同的地址。 |
| principalAmount  （本金数量） | uint256 | 债务人请求的本金总数量。 |
| principalToken  （本金通证地址） | address | 本金代币的ERC20合约地址。 |
| debtorFee  （债务人手续费） | uint256 | 将从债务人的本金中扣除的总手续费。请注意，债务人和债权人支付的费用总额必须等于支付给承销商和转包商的费用总额。 |
| creditorFee  （债权人手续费） | uint256 | 债权人在本金基础上需支付的额外手续费。请注意，债务人和债权人支付的总手续费应等于付给承销商和转包商的全部费用。 |
| Relayer  （转包商） | address | 列出了给定债务但得转包商地址。 |
| relayerFee  （转包商手续费） | uint256 | 当债务人与债权人的关系最终确定时，债务核心将支付转包商的手续费。 |
| underwriterFee  （承销商手续费） | uint256 | 当债务人与债权人的关系最终确定时，债务核心将支付承销商的手续费。 |
| Expiration  Timestamp  （过期时间） | uint256 | 超过这个时间点交易或订单还未成交则失效。 |

### 承销商承诺哈希 Underwriter Commitment Hash

承销商承诺哈希是承销商签名的有效数据，以表明他同意债务订单上的参数。承销商需要签一个不同的参数子集，这样债务人不需要向每个为他们广播其债务订单的转包商点申请新的签名。我们将在下面进一步阐述。

| **字段** | **数据类型** | **描述** |
| --- | --- | --- |
| kernelVersion  （核心版本） | address | 债务核心合同的地址。协议升级时，这个地址将被更新。 |
| issuanceHash  （发行哈希） | bytes32 | 与此发行承诺关联的还款路由器合同的地址。 |
| principalAmount  （本金数量） | uint256 | 债务人请求的本金总数量。 |
| principalToken  （本金通证地址） | address | 本金代币的ERC20合约地址。 |
| underwriterFee  （承销商手续费） | uint256 | 当债务人与债权人的关系最终确定时，债务核心将支付承销商的手续费。 |
| Expiration  Timestamp  （过期时间） | uint256 | 超过这个时间点交易或订单还未成交则失效。 |

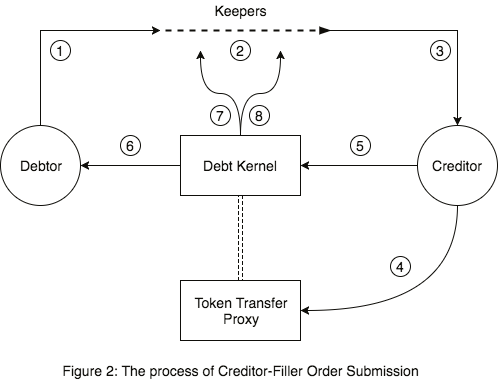
总的来说，如果以下签名均已附上，债务订单会被认为已准备好提交:

1. 债务人对债务人/债权人承诺哈希的ECDSA签名。(除非债务人向达摩智能合约提交订单，否则必须提交)；
2. 承销商对承销商承诺哈希的ECDSA签名。(如果没有承销商在场，或向达摩智能合约提交订单的地址为该承销商，则不需要)；
3. 债权人对债务人/债权人承诺哈希的ECDSA签名。(除非向达摩智能合约提交订单的地址为该债权人，否则必须提供)；

### 债务发行流程 Debt Issuance Process

拥有有效的债务订单并附带必要的附加签名的任意一方，都可将该订单提交至达摩智能合约以填写订单。将债务订单提交到达摩智能合约后，一个债务协议通证将被发行并与本金同步交换。我们将债务人完成的债务订单称为“**债务人完成订单的提交**”（Debtor-Filler Order Submissions），并将债权人完成的债务订单称为“**债权人完成订单的提交**”（Creditor-Filler Order Submissions）。

### 债权人完成订单的提交 Debtor-Filler Order Submissions



以下步骤与上图中的圈中数字相对应:

1.债务人向承销商申请贷款。

2.在债务人、承销商和转包商间将发生债务订单握手（下文将详细描述），随后转包商将列出一个有效的、完整的债务订单。

3.债权人对转包商维护的公共订单薄上的债务订单进行评估。

4.如果债权人想要完成订单，他首先会授权通证转让代理（token transfer proxy）可以转让大于等于本金加上债权人费用的代币（即使用ERC20的approve函数）。请注意，此步骤不需要在债权人填写的每一个订单中重复，债权人可以向通证转移代理进行一次授权，使该代理有权限转移很大数量的通证。只有当他通过提交或为签一份债务订单的时候，智能合约才会从他的帐户中取出相应的代币通证。

5.然后债权人将其直接提交给债务核心合约。请注意，在这种情况不需要他的签名，因为他向核心提交的订单就已经代表了他同意其参数。债务核心接着向债权人发放一个非同质性、不可分割的通证，代表债务人对合同和相关参数的承诺。

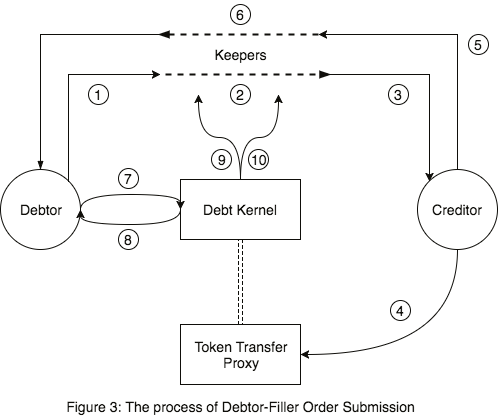
6.债务核心将数量为本金-债务人手续费的贷款转入债务人账户。

7.债务核心根据债务订单中定义的承销商手续费转入承销商账户。

8.债务核心根据在债务订单中定义的转包商手续费转入转包商账户。

债务人制定计划对有许多潜在债权人有利，债务人并不关心去控制债务最终发行的确切时刻。大多数情况下，债务人制定计划可能最有效。

### 债务人完成订单的提交 Creditor-Filler Order Submissions



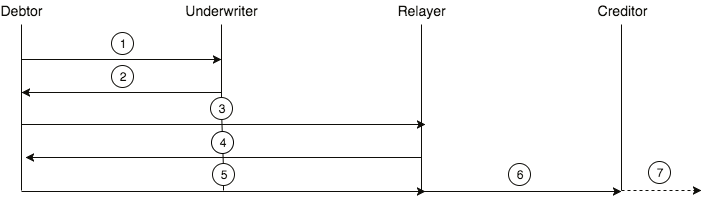
以下步骤与上图中的圈中数字相对应:

1. 债务人向承销商申请贷款。
2. 在债务人、承销商和转包商间将发生债务订单握手（下文将详细描述），随后转包商列出一个有效的、完整的债务订单。
3. 债权人对转包商维护的公共订单薄上的债务订单进行评估。
4. 如果债权人想要完成订单，他首先会授权代币转让通证（token transfer proxy）可以转让大于等于本金加上债权人费用的代币（即使用ERC20的approve函数）。然后，在债务订单的债务/债权人承诺协议里附上其ECDSA签名。
5. 债权人将已签署的债务订单提交给转包商。
6. 转包商将已签署的债务订单转发给债务人。
7. 债务人在方便的时候将签署的债务订单提交给债务核心，这将发行一个独特的、非同质性债务订单通证给债权人，使其成为债务订单的最终签字人。
8. 债务核心将数量为本金-债务人手续费的贷款转入债务人地址。
9. 债务核心根据在债务订单中定义的承销商手续费转入承销商地址。
10. 债务核心根据在债务订单中定义的转包商手续费转入转包商地址。

债务人完成方案对于债务人希望同步地借用通证作为另一个更大的交易的一部分这种情景是十分有用的。例如，如果一个智能合约要求用户支付一定数量用于存储的通证(如FileCoin、Storj等) 以使用它，用户可以通过上述方案获得一个有效的、已签名的债务订单来作为智能合约函数调用的一个参数。然后，智能合约可以将债务人的订单提交给债务核心，同时向债务人提供必要的存储通证，然后在一笔交易中将其记入智能合约。这大大减少了在几乎任何情况下信用借贷情景里执行交易的摩擦。

### 债务订单握手流程 Debt Order Handshake

上面提到的债务订单握手流程定义如下：



1. 债务人从指定的承销商处请求其承销的债务，列举他想要的贷款条件（即本金类型、期限等）。
2. 承销商使用专有风险模型评估债务人的违约风险，构建承销商承诺(同时需要构建债务发行承诺)，将其ECDSA签名附到承销商承诺的哈希上，并将债务发行承诺、承销商承诺和ECDSA签名发送给债务人。
3. 如果参数符合债务人的期望条件，债务人现在便拥有了足以构建一个完整的债务订单的所有必要参数。但如果他希望将订单转给潜在的债权人，债务人将需要一个转包商费用计划和地址。
4. 转包商以他的费用计划和地址回应债务人的请求。
5. 如果转包商的费用符合债务人的意愿，债务人将使用承销商和转包商提供的参数构建一个完整的债务订单，并将订单发送给转包商。
6. 转包商在他们的订单簿上列出完整的债务订单。鉴于他们自愿列出订单这一事实，他们对订单参数的同意得到了默认，故其签名不是必要的。

### 债务偿还流程 Debt Repayment Process

为了对任何债务资产的偿还状况都能进行实证评估，我们定义了一个还款流程，在还款流程中，由还款路由合同引导完成，整个过程将不可修改地记录在链上。

当债务人希望偿还债务时，他们将执行以下操作:

1. 债务人会授予通证转让代理权(即通过ERC20 approve 函数)，其数额大于或等于所期望的偿还金额。
2. 债务人将发送交易到还款路由合约，附带上参数确定的所需还款金额以调用偿还功能。然后，还款路由合约将获取该债务的当前受益人的地址(即债务代币的持有者)，将所期望的还款金额从债务人的帐户转移到受益人的帐户上，通过调用债务的条款合同中的 registerRepayment 方法登记该笔还款。

注意：我们可以构建一个更简单的方案，在这个方案中，债务人直接向债权人偿还债务，而不需要利用债务核心合约。然而，还是有必要让对于债务核心促成过程，以保证当一笔还款在特定的条款合同中登记时，有一笔还款交易与之对应。

### 条款合约接口 Terms Contract Interface

我们要求通过达摩协议所发行的任何债务都承诺签订一份“智能合约”，即条款合约。条款合约的作用是提供一个不可变的、程可序化查询的、可证明的可信来源，来揭示债务的偿还状态。这使我们能够以经验和明确的方式在债务发行过程中定义还款计划，并在债务的生命周期中对债务的偿还状况进行评估。所需要的功能接口如下：

/// 条款合约接口

interface TermsContract {

/// 当被调用时，registerRepayment 函数登记债务人的偿还交易，并附带上一些元信息，如对美元汇率等。

/// @param agreementId bytes32. 本协议所涉及的债务协议的协议ID（发行哈希）

/// @param payer address. 付款者地址

/// @param beneficiary address. 付款受益人的地址

/// @param unitsOfRepayment uint. 交易中偿还的价值单位

/// @param tokenAddress address. 执行还款交易的令牌的地址

function registerRepayment(

bytes32 agreementId,

address payer,

address beneficiary,

uint256 unitsOfRepayment,

address tokenAddress

) public returns (bool \_success);

/// 一个registerRepayment的变种，将还款信息记录在NFT代币中。

/// 事后确定偿还的价值（例如当前美元汇率）

/// @param agreementId bytes32. 本协议所涉及的债务协议的协议ID（发行哈希）

/// @param payer address. 付款者地址

/// @param beneficiary address. 付款受益人的地址

/// @param tokenId 支付交易中NEFT转移的令牌ID

/// @param tokenAddress 执行还款交易的令牌的地址

function registerNFTRepayment(

bytes32 agreementId,

address payer,

address beneficiary,

uint256 tokenId,

address tokenAddress

) public returns (bool \_success);

/// 返回到特定区块高度时，累计需要偿还的价值单位。

/// 注意这不是一个常量函数，函数的返回值对于给定的区块高度也可能发生改变，因为利率可能会在交易几方间被重新协调。

/// @param agreementId bytes32. 本协议所涉及的债务协议的协议ID（发行哈希）

/// @param blockNumber uint. 正在查询还款期望的block编号

/// @return uint256 预期在特定的blockNumber失效时偿还的累计价值单位

function getExpectedRepaymentValue(

bytes32 agreementId,

uint256 blockNumber

) public view returns (uint256);

/// 返回到特定区块高度时，累计已偿还的价值单位。

/// @param agreementId bytes32. 本协议所涉及的债务协议的协议ID（发行哈希）

/// @param blockNumber uint. 正在查询还款值的block number

/// @return uint256 在给定的blockNumber失效时偿还的累计值单位。

function getValueRepaid(

bytes32 agreementId,

uint256 blockNumber

) public view returns (uint256);

}

请注意，在getExpectedRepaymentValue和getValueRepaid函数中，还款数量被抽象地定义为了“价值单位”。 这个单位是被有意义的留下来不作定义，这使得债券发行人有了更多的灵活性。比如说，在执行代币的实际交易时，可以用法定货币来表示预期的偿还价值。

### 违约和收债 Defaults and Collections

达摩协议对承销商处理违约的手段和回收债务的方式不加限制。表面上，一些承销商可以与债务人发行具有法律约束力的贷款协议，并通过法院利用债务收取债务。当然也可以使用链上抵押机制，利用特定债务的承诺条款合约所提供的功能，只要当前价值偿还单位低于预期，就无偿地向债权人发放抵押品。

无数其他的方案可以被用来抑制违约，**达摩协议不会提倡或设计任何特定的解决方案，而是旨在提供一种标准机制，通过这种机制来研究承销商对于债务资产的历史评级**，以评估承销商的表现。市场应该倾向于奖励那些过去表现强劲的承销商，反之亦然去惩罚过去表现欠佳的承销商。承销商过去绩效的衡量标准，我们称之为*Fβ*Fβ，它借鉴了统计分析中二元分类的统计分析，目的是对承销商违约预测的准确程度做分类：

定义：x ∈ 1,...,n 是承销商在投资组合中评级的债务

定义：α\_x 是债务x合约到期时，债务人期望偿还的总价值。

定义：γ\_x 是债务x合约到期时，债务人实际偿还的总价值。

定义：δ\_x 是在承销商的预测中，债务x违约的可能性。

定义：β为调整召回率对准确性重要程度的超参数。

p=∑{x}min(α\_x−γ\_x,δ\_x\*α\_x)/∑{x}δ\_x\*α\_x

r=∑xmin(α\_x−γ\_x,δ\_x\*α\_x)/∑\_x(α\_x−γ\_x)

F\_β=(1+β^2)\*p\*r/β^2\*p+r

**在这强调这不是一个对承销商的表现进行评估时可以用到完全可靠且包罗万象的度量指标。恶意欺诈的承销商可以通过各种方式(参见"攻击"小节**)来戏耍该指标。应该说这是一个经验信号，善意的、可信的承销商可以被透明地评估。当市场上的承销商都值得信赖时，该指标将是一个非常有价值的信号。

### 用例

债务是一种非常多样化的资产类别，理论上，几乎任何类型的债务协议都可以通过达摩协议来发行和承销。 任何类型法币借贷业务的在线放贷机构通过在每笔交易中充当债务人和承销商，都可以将他们的后台迁移到达摩协议上。将本金支付转化为收据上的法币,并使用法币进行所有付款和借款，从而对用户完全隐藏那些贷款流程链上部分的复杂细节。我们认为，这最终会为寻求债务资本的在线贷款机构提供一个有吸引力的替代途径。

然而，在短期内，链上债务发行的更具有代表性的案例将是那些仅能在链上实现或因为区块链得到拓展的场景。下面我们将重点介绍其中的几个:

### 首次债券发行/通证化SAFTs

将ICO放在一起是需要惊人的努力，这个领域的项目开始不断增加其pre-ICO轮的募资数额以完成其最终的ICO轮。这促成了SAFT[6]的诞生：SAFT是一种有效运作类似于可转换债务协议的法律工具。表明投资者出资X美元，在未来的一笔通证筹集中，它希望以折扣价获得价值X美元的通证。

我们建议签发一种基于密码学协议的pre-ICO融资条款，可发行可筹集可被标记化，从使用条款的角度来看，操作起来将比现有的私募轮更灵活。

在最简单的例子中，债务条款合同可以规定一种类似于SAFT的普通折扣预售协议。借出的原则是提出本金X美元，预期的还款是价值X美元的最终以折扣价出售的价值通证。因此，投资者将得到一种可转换债券，表明他们在最终的价值通证中有折价所有权。然而，这种安排的好处不仅仅是简单在代理通证中出售通证，而是最终会成为价值通证。

当合同条款可以编码各种各样的投资者补偿方案时，这些安排将变得更加有用和有趣。例如，考虑预售时，债务人在最终ICO时，已筹集资金的90%,$X,使用以太币以Y%的利率偿还，其余的10%以有折扣的价值通证支付。这类似于传统金融体系中的风险债务融资，并将给投资者提供一个风险的下限(即，90%的X美元押注于ICO的成功)，而上升潜力也降低(即X美元的10%将押注于该价值通证的最终市场需求)。这可以通过创建一份合约来进一步增强，该合约作为合约的多重拥有者之一，在最终的ICO中收取收益，将收益锁定为抵押，直到债务条款合同表明预售投资者得到了适当的补偿。更多的预售方案可以使用债务条款合同在达摩协议中实现（例如，不同的折扣方案会基于债务人达到最终ICO的时长制定。如果一个ICO未能成功募集到目标资金，返还投资人资金的方案等等）。

### 去中心化的杠杆借贷 Decentralized Margin Lending

依赖于链下订单薄（如0x协议）的去中心化交易所是代币生态系统中重要而必要的基础设施。 但是，如果他们想要在实用性和用户体验方面与中心化的交易所寻找合适的交易，他们需要以某种形式支持保证金交易。

我们可以通过达摩协议发放保证金贷款来构建一个去中心化保证金借贷系统，并将合同上的条款与指定的债务作为一种闸门函数，允许代理人锁定在智能合约中的保证金。 在这样的计划中，承销商将要求债务人根据预期的保证金比率将抵押品锁定到保证金账户合约中，作为对借款人的债务发行承诺的先决条件。 然后，债务人将在一笔保证金交易中同步地完成债务订单和0x的广播订单，收款账户为被锁定的保证金账户。持仓期间，承销人将作为条款合同提供可信的价格指数预言机。关闭帐户将属于债权人的职权范围，如果合同条款表明出现了保证金帐户处于违约状态(按约定的保证金比率)或者仓位已经过期的情况，此时合同保证金帐户将赋予债权人关闭账户和获得她想要利息的权利。

基因此，我们可以利用该协议建立一个保证金贷款系统，该系统可以在现有的去中心化交易所中与很自然的与各类转包商提供的订单簿集成。

## 攻击和局限性

### 欺诈的承销商 Fraudulent Underwriters

以欺诈方式运作的承销商可以欺骗投资者，让他们相信原始债务主要来自借款人，但他们自己实际上就是借款人。 可以想象，一个欺诈性的承销商可以通过使用许多虚假的公钥向自己贷款，并以此来建立一个极其优质的评级表现历史。然后通过欺骗债权人来投资他们的债务订单，在最后来执行大规模推出，其中他们是真正的借方和完全拖欠债务。 正是由于这个原因，我们强调，达摩并不是一个完全无需信任的协议——在这个方案，借贷和放贷只有在承销商是已知的、可信的而不仅是一个匿名公钥的情况下才安全。 所以，尽可能建立债权人对自己的信任是承销商的责任。为了实现这一点，承销商既可以通过依托已有的公开确定的身份实体的信誉(例如，债务人是一个企业或公司)，或通过培育公开可信并且已知的与监管机构关系(例如， 大多数现有的在线贷款提供商参与该协议)。 承销商必须与投资方建立信任，这与一个协议的开发团队进行一轮募资的过程是一样的。两者都在促成公共投资项目的发行，因此，两者都有责任维护自己合法性和正直性。

### 女巫攻击 Sybil Attacks

在贷款协议的背景下，女巫攻击指使用大量的欺诈身份，同时承担多个无担保债务，并且一次性违约，几乎没有任何后果。在很大程度上，防御女巫攻击是承销商的职责——如果承销商的身份证明和KYC程序不够强大，不足以检测欺诈身份的行为，们过往的评级表现中会反映出来。然而，女巫攻击者可能会在一次性内从几个承销商那里同时请求债务，这意味着在类似借贷领域的承销商之间的数据共享很可能会成为重要的需求。

### 债权人恶意破坏 Debtor Griefing

在交易中债权人扮演债务订单做市商的情况下，债权人表面上可以实施我们称之为“债务人恶意破坏”的攻击中。债权人为了填补另一份订单，提前将锁定在另一份，即将交易合约中的本金提走。 为了执行攻击，债权人将在握手过程中作为订单做市商，协助提交一笔债务订单，等待债务人的完成交易，通过监视当地的以太坊节点的待定交易池，在同一个区块中提交交易，将他的通证从本金代币所在的账户移出，从而导致债务人的交易失败并抛出异常。 这样的攻击会导致债务人完成订单的交易抛出异常，并使得债务人浪费少量gas手续费。 在简单的一次性交易情景中，最坏的情况就是浪费gas手续费。 然而，对有组织地攻击的情况，债权人显然可以使用这种机制来对一个转包商的整个订单簿进行拒绝服务的攻击，因为他是一个做市商（订单提交者），利用连续生成的地址，在订单簿中，不断地对债务订单进行处理，不断地对债务人进行诈骗，并回收资金用作于进一步的恶意攻击。 大规模执行此类攻击的gas成本是不可忽视的，但是，假设攻击者的动机是来自转包商对于其他转包商的某种敲诈勒索，如果攻击者拥有必要的流动性，攻击就会被证明是有利可图的。预防此类攻击的最佳防线是转包商自己——受到此类攻击的转包商有各种各样的选择可以缓和恶意债权人的拒绝服务攻击，从简单的IP拦截，到更激进的政策如要求债权人进行KYC流程检查。

### 私底交易 Side Deals

我们称“私底交易”指的是在特定债务交易中，代理人试图避免向组成交易的监管员支付费用。 考虑以下例子，一个第三方维护的在线注册表，潜在债务人自愿把他们的公钥和他们的邮箱地址相关联。 债权人可以在浏览器上查看转包商订单簿上的债务订单，通过比对订单薄上的公钥，债权人可以利用第三方的在线注册表来债务人进行沟通。 然后，债权人和债务人可以在另一个渠道上沟通，建立一个点对点的债务订单，在这种订单中，承销商和转包商都不能收到手续费，从而节省了相关的费用。

承销商对这种漏洞可能不那么敏感，因为债权人在整个债务期限内都要依赖他们的服务。 然而对于转包商来说，债权人进行此类攻击时并不会有任何损失。这意味着，对手通过构建一个公共可用的匹配引擎来系统性地破坏一个转包商的操作是可行的，它可以从中继者的订单簿中获取债务订单，并匹配参与的债务人和债权人，这样他们就可以建立一个交易。

我们认为，在实际中，参与系统性私底交易匹配对于代理具有相当高的摩擦成本，因为成功的交易匹配将需要代理参与另一个债务订单握手。在高度异步的情况，会耗费大量的时间才能节省相关的费用。 但即便我们构建了一个可以使用的没有摩擦成本的私下、底交易匹配机制，转包商也不是就没有追索权了。 有针对性的转包商可以通过一个“提交-执行”二步结构来混淆债务人的地址，而这一结构可以保留转包商与债权人之间的信任关系[7]:

1. 除开往常列出债务订单，转包商可以列出被修改过的债务订单，比如隐藏包含债务人地址的字段都，包括相关的信息及其签名。仅提供能让债权人决定是否要放贷的信息点。
2. 提交：债权人通过将无效和不完整的债务订单提交给专门建立的智能合约来填写订单，承诺会完成订单，该合约在交易中充当中间人。此外，债权人授予该智能合约以一定数额的转让额(allowance)[8]，相当于附上0x广播用通证来最终完成该订单。
3. 执行：转包商通过提交完整的、有效的债务订单来执行上述合约的发行和交换。然后，该合约验证所提交的债务订单中的所有字段与债权人提交的、不完整的债务订单(除了包含债务人地址的字段外)对应的所有字段，然后依次从债权人转移其所需的本金通证，作为中间债权人向债务核心合约提交完整的债务订单，然将其新拥有的非同质化债务通证转让给债权人。如果提交的债务订单无效或与债权人承诺的债务订单不匹配，则合同抛出异常，债权人继续持有其债务通证。

因此，如果情况需要，转包商可以利用上面的“提交-执行”结构来给债权人以信任的方式来完成被混淆过的债务订单。

## 脚注

1. 这里强调的是“类似股票”的，在理论上，协议代币不是股权，但是，就他们的风险概况和他们所吸引的投机兴趣的类别而言，其表现出来的行为就像股票一样。 ↩
2. Warren, Will, and Amir Bandeali. “0x: An Open Protocol for Decentralized Exchange on the Ethereum Blockchain.”

<https://0xproject.Com>， 0xproject.com/pdfs/0x\_white\_paper.pdf.

1. Zurrer, Ryan. “Keepers - Workers That Maintain Blockchain Networks.” Medium, Medium,5 Aug. 2017，

medium.com/@rzurrer/keepers-workers-that-maintain-blockchain-networks-a401826 15b66.

1. 在早先版本的白皮书中，我们将这些实体称作”风险评估者“，或者简写成”RAA“。这个版本中，我们去掉了这些术语，转向选用更容易理解的表达方式：承销商。 ↩
2. Grigg, Ian. “The Ricardian Contract.” Iang.org, 2000, iang.org/papers/ricardian\_contract.html ↩
3. Batiz-Benet, Juan, et al. “The SAFT Project: Toward a Compliant Token Sale Framework.” saftproject.com/static/SAFT-Project-Whitepaper.pdf. ↩
4. 注意这种构建仅限于债务人构建的债务订单。 ↩
5. 通过ERC20的approve方法。 ↩